

UNIVERSIDAD CARLOS III DE MADRID  
ESCUELA POLITÉCNICA SUPERIOR  
GRADO EN INGENIERÍA INFORMÁTICA CON MENCIÓN EN  
COMPUTACIÓN

# Desarrollo de una herramienta software para la extracción de datos sobre el rendimiento de la red eléctrica

Alumno: Jorge Carlos López Hornero  
Tutor: Rosa María Romero Gómez  
06/10/2015

## Agradecimientos

---

A mi madre Rosa y a mi padre Carlos, por su continuo e incesante apoyo, por su ayuda en los malos momentos que hacen dudar y por compartir los buenos momentos que ayudan a seguir adelante. Por aportarme su experiencia en los estudios, en el trabajo y sobre todo en la vida. Por transmitirme sus valores y sus cualidades, por no sólo educar sino por algo tan complicado e importante como es enseñar. Sin su ayuda el viaje desde la primera asignatura hasta este proyecto final habría sido simplemente imposible.

A mis hermanos Sandra y Álvaro y también a Cova, porque ellos antes que yo emprendieron el mismo viaje con éxito y han sabido ayudarme y aconsejarme durante mi recorrido de forma constante, pues aunque siempre se dé por hecho a un hermano, ellos lo han hecho patente.

A mis abuelos Amado, Ana, Emilio y Rosa. A los que están y a los que ya se fueron, pues ellos mejor que nadie han sabido celebrar cada uno de mis triunfos y exhiben con orgullo la foto de la orla de su nieto. Por su aprecio incalculable, por su tiempo y su sonrisa constante reflejada en su cara independientemente de lo que pasara por su cabeza. Por esa actitud que recuerdo desde siempre que consigue que los problemas pasen de ser una montaña a un granito de arena.

A Sergio, ese hermano con el que a pesar de no compartir padres ha demostrado que el concepto de hermandad está por encima de los lazos de sangre. Por su constante apoyo, su ayuda y paciencia infinitas, a las que quizá haya puesto a prueba demasiadas veces. Por estar siempre disponible, sufrir las desgracias como si fueran propias, compartir las alegrías y celebrar los triunfos, él es simplemente, parte de mi familia.

A Juancar, por quitarme el miedo a cambiar de etapa, por esas prácticas interminables y todas esas asignaturas compartidas, por su ayuda y su apoyo, siempre presente con su interminable optimismo en los buenos y los malos momentos y quizá nunca lo suficientemente valorado. Por demostrar que por encima de los cabreos, el desgaste de las entregas infinitas y demás nimiedades, queda la hermandad y la amistad.

A todos mis amigos, los viejos, los nuevos y los que se han quedado por el camino, pues todos ellos forman parte de mí y me han formado como persona. Por alegrarme y animarme en los malos momentos, por distraerme, por aguantarme cuando el estrés me superaba. Por lo que han hecho y lo que sin duda seguirán haciendo, espero poder devolvérselo con creces.

A mis compañeros de trabajo y en especial a mis compañeros de departamento, pues gracias a ellos cada día es ameno y distinto a pesar de la rutina diaria típica de la oficina, ellos consiguen en definitiva que no parezca un lugar de trabajo.

A mi tutora Rosa, por conseguir guiarme y aguantarme a través de todo este proceso. Por sus constantes correcciones y aportaciones, por su experiencia, por conseguir hacerme ver la luz al final del túnel y llevar este trabajo a buen puerto. Por hacer que lo que parecía un logro imposible de alcanzar, se convierta hoy en un trabajo terminado.

A todos vosotros, porque todos estos años habéis estado presentes en mi vida de una forma u otra, este trabajo es tan vuestro como mío. ¡Gracias a todos!

## Resumen

---

El presente trabajo fin de grado se enmarca dentro del contexto de las herramientas software para la extracción de datos o “data scraping”, lo que en castellano podría traducirse como “rascado” o “raspado” de información. Esta extracción de información a través de herramientas software o “data scrapers” se realiza mediante la aplicación de diversas técnicas en las que un ordenador extrae información legible por el ser humano a partir de un conjunto de datos en bruto. Estos data scrapers se pueden dividir en dos tipos en función de la fuente de la información utilizada, los “screen scrapers”, los cuáles extraen información de una fuente visual como puede ser un terminal de ordenador y los “web scrapers”, los cuáles utilizan los lenguajes de marcado como HTML para obtener información. Es en este segundo tipo de “data scraper” en el cual se focaliza este trabajo. Esto se debe a la necesidad dada por el departamento de Ingeniería Eléctrica de la Universidad Carlos III de Madrid de extraer información sobre el consumo eléctrico desde la página web de Red Eléctrica de España para su posterior análisis.

El consumo eléctrico es una constante en nuestra vida diaria, tanto para nuestro uso personal como indirectamente para la industria o el sector servicios. Dicho consumo, al producirse y contabilizarse, genera datos que son acumulados y pueden ser estudiados con distintos propósitos. El gran obstáculo de este proceso reside en la dificultad de la extracción de estos datos de forma masiva y estructurada, mediante los medios proporcionados por Red Eléctrica de España. A fin de solventar esta problemática, el presente trabajo fin de grado provee una herramienta software que logra la extracción y agregación eficiente de grandes volúmenes de datos sobre el rendimiento de la red eléctrica y su exportación a archivos pretratados para su posterior uso en herramientas de cálculo, así como su visualización.

La evaluación del cubrimiento de los objetivos definidos en el trabajo se ha realizado a través de la realización de distintos casos de prueba. Adicionalmente, se ha realizado un cuestionario de usabilidad a un grupo cerrado de diez usuarios a fin de determinar la facilidad de uso y aprendizaje de la herramienta.

## Abstract

---

This undergraduate final project is framed within the context of “data scraping”, which is a technique in which a computer extracts data from human-readable output coming from another program. Accordingly, data scrapers are software tools designed to collect data from different sources in an automated way. They can be divided into two types depending on the source, “screen scrapers”, which are associated to the collection of visual data from a source such as a computer display, and “web scrapers”, which use text-based mark-up languages such as HTML to extract data. This project is focused on the latter given the need of the Electrical Engineering Department of Carlos III University of extracting power consumption-related data from the Red Eléctrica de España public website in order to analyze it.

Power consumption is a constant in our daily lives, both for our personal use and, indirectly, for the industry and services sector. This consumption is recorded while is being produced and counted, which can be used afterwards for different purposes. The major obstacle to this process is the difficulty of extracting this power consumption-related data in bulk and in a structured way from the Red Eléctrica de España public website. To solve this problem, this final project provides a software tool that not only manages the aggregation, extraction, and export of large volumes of data for later use in calculation tools but also the visualization of such data.

The evaluation of the objectives defined in this project has been carried out through conduction of different test cases. In addition to that, a usability testing with ten users has been also conducted to determine the ease of use and learning of the proposed software tool.

## Índice de contenidos

Agradecimientos .....	2
Resumen.....	3
Abstract .....	4
Índice de contenidos .....	5
Índice de figuras .....	6
Índice de tablas .....	8
Glosario de términos.....	11
1    Introducción.....	13
1.1    Definición del problema .....	14
1.2    Objetivos .....	14
1.3    Fases del desarrollo del proyecto .....	15
1.4    Medios empleados.....	15
1.5    Estructura de la memoria.....	16
2    El estado de la cuestión .....	17
2.1    La extracción de datos o “data scraping” .....	17
2.2    Elección de las tecnologías de desarrollo.....	26
3    Gestión de proyecto software .....	33
3.1    Definición del proyecto .....	33
3.2    Estimación de tareas y recursos.....	33
3.3    Presupuesto .....	36
4    Solución.....	41
4.1    Descripción de la solución.....	41
4.2    El proceso de desarrollo.....	43
5    Evaluación .....	94
5.1    Proceso de evaluación.....	94
5.2    Análisis de resultados.....	101
6    Conclusión.....	107
6.1    Propósito del trabajo y aportaciones realizadas.....	107
6.2    Trabajos futuros .....	107
6.3    Problemas encontrados .....	108
6.4    Opiniones personales.....	108
7    Bibliografía .....	110
Anexo I. Control de versiones .....	113
Anexo II. Diagrama del desarrollo de la herramienta .....	114
Anexo III. Entrevistas con el cliente .....	115
Anexo IV. Manual de usuario .....	116
Anexo V. Prototipo .....	121
Anexo VI. Formulario de usabilidad .....	128

## Índice de figuras

Figura 1: Curva de la demanda de un día normal .....	13
Figura 2: Soluciones de la gestión de la demanda .....	13
Figura 3: Proceso de obtención de los datos para todo un año .....	14
Figura 4: Ejemplo de uso de Sikuli.....	18
Figura 5: TeamViewer en plena conexión.....	19
Figura 6: WebHarvy obteniendo información de la web de Páginas Amarillas .....	21
Figura 7: Pantalla principal de Web Data Extractor .....	22
Figura 8: Página principal de import.io .....	23
Figura 9: iMacros extrayendo información en Chrome .....	24
Figura 10: Vista de la web de Red Eléctrica desde Firebug para Firefox.....	24
Figura 11: Ejemplo de uso de WebScrapers en Chrome .....	25
Figura 12: Ejemplo de uso de cURL en consola.....	25
Figura 13: Los diez lenguajes de programación más usados .....	29
Figura 14: Interfaz sencilla en Swing .....	30
Figura 15: JavaFX Scene Builder con uno de los prototipos de este proyecto.....	31
Figura 16: Esquema de la solución.....	41
Figura 17: Interfaz de usuario de la aplicación tras realizar una petición simple .....	42
Figura 18: Ciclo de vida tradicional con retroalimentación .....	44
Figura 19: Esquema de la arquitectura Modelo Vista Controlador.....	61
Figura 20: Perspectiva de bienvenida .....	86
Figura 21: Perspectiva de respuesta .....	87
Figura 22: Entorno de desarrollo Eclipse con el proyecto abierto.....	88
Figura 23: JavaFX Scene Builder con un archivo del proyecto abierto .....	88
Figura 24: Asistente de creación de proyectos de JavaFX en Eclipse.....	89
Figura 25: Paquetes que componen el proyecto .....	89
Figura 26: Un archivo de vista y su respectiva clase controladora .....	90
Figura 27: Diálogo de error creado con ControlsFX .....	90
Figura 28: Estructura de carpetas del proyecto .....	91
Figura 29: Contenido de la carpeta resources .....	91
Figura 30: Contenido de la carpeta build .....	91
Figura 31: Botón de la interfaz modificado con hojas de estilo.....	92
Figura 32: Ejemplo de archivo FXML y su representación .....	92
Figura 33: Ejemplo de informe exportado compatible con MatLab .....	93
Figura 34: Experiencia de los usuarios encuestados.....	102
Figura 35: Resultados de la primera pregunta.....	102
Figura 36: Resultados de la segunda pregunta .....	103
Figura 37: Resultados de la tercera pregunta .....	103
Figura 38: Resultados de la cuarta pregunta .....	104
Figura 39: Resultados de la quinta pregunta .....	104
Figura 40: Resultados de la sexta pregunta .....	104
Figura 41: Resultados de la séptima pregunta.....	105
Figura 42: Resultados de la octava pregunta .....	105
Figura 43: Resultados de la novena pregunta.....	106
Figura 44: Resultados de la décima pregunta .....	106
Figura 45: Diagrama de Gantt de la última iteración .....	114

Figura 46: Sección de bienvenida.....	117
Figura 47: Sección de generación de informes .....	117
Figura 48: Parámetros del informe .....	118
Figura 49: Apariencia de la aplicación tras realizar el informe .....	118
Figura 50: Visualización del informe .....	119
Figura 51: Cambio de informe en petición por intervalos .....	119
Figura 52: Pantalla de guardado del sistema .....	120
Figura 53: Ejemplo del mockup del programa .....	121
Figura 54: Toma de contacto con el problema .....	121
Figura 55: Primer intento de interfaz con JavaFX .....	122
Figura 56: Primera interfaz semifuncional .....	122
Figura 57: Primera acercamiento al resultado final .....	123
Figura 58: Resultados de la consulta mostrados en la interfaz.....	123
Figura 59: Añadida la posibilidad de realizar peticiones por rangos de fecha.....	124
Figura 60: Recolocación de los botones y adición de la fecha actual .....	124
Figura 61: Cambio de los campos de texto por tablas y otras mejoras .....	125
Figura 62: Añadida la gráfica y otras mejoras .....	125
Figura 63: Ejemplo de paginación .....	126
Figura 64: Ejemplo de sincronización de desplazamiento vertical entre tablas .....	126
Figura 65: Ejemplo de Oracle de gráfica de barras apiladas .....	127
Figura 66: Formulario de usabilidad de la aplicación.....	129

## Índice de tablas

Tabla 1: Acrónimos.....	11
Tabla 2: Términos.....	12
Tabla 3: Comparativa de tipo de aplicaciones .....	27
Tabla 4: Comparativa de lenguajes de programación.....	30
Tabla 5: Comparativa de interfaces para Java .....	32
Tabla 6: Tareas del proyecto y sus responsables .....	34
Tabla 7: Ventajas de las metodologías ágiles frente a las tradicionales .....	36
Tabla 8: Tareas del proyecto y su duración aproximada .....	36
Tabla 9: Salarios brutos medios por profesión .....	37
Tabla 10: Salario bruto mensual por trabajador .....	37
Tabla 11: Bases de cotización por trabajador .....	38
Tabla 12: Coste total mensual por trabajador .....	38
Tabla 13: Coste total del personal.....	38
Tabla 14: Coste de hardware .....	39
Tabla 15: Coste de consumibles.....	39
Tabla 16: Coste de material total sin IVA .....	39
Tabla 17: Coste total bruto del proyecto .....	40
Tabla 18: Coste final del proyecto.....	40
Tabla 19: Plantilla para los requisitos funcionales y no funcionales.....	45
Tabla 20: Requisito funcional RF-001.....	46
Tabla 21: Requisito funcional RF-002.....	46
Tabla 22: Requisito funcional RF-003.....	46
Tabla 23: Requisito funcional RF-004.....	46
Tabla 24: Requisito funcional RF-005.....	47
Tabla 25: Requisito funcional RF-006.....	47
Tabla 26: Requisito funcional RF-007.....	47
Tabla 27: Requisito funcional RF-008.....	47
Tabla 28: Requisito funcional RF-009.....	47
Tabla 29: Requisito funcional RF-010.....	47
Tabla 30: Requisito funcional RF-011.....	48
Tabla 31: Requisito funcional RF-012.....	48
Tabla 32: Requisito funcional RF-013.....	48
Tabla 33: Requisito funcional RF-014.....	48
Tabla 34: Requisito funcional RF-015.....	48
Tabla 35: Requisito funcional RF-016.....	49
Tabla 36: Requisito funcional RF-017.....	49
Tabla 37: Requisito funcional RF-018.....	49
Tabla 38: Requisito funcional RF-019.....	49
Tabla 39: Requisito funcional RF-020.....	49
Tabla 40: Requisito funcional RF-021.....	49
Tabla 41: Requisito funcional RF-022.....	50
Tabla 42: Requisito funcional RF-023.....	50
Tabla 43: Requisito funcional RF-024.....	50
Tabla 44: Requisito funcional RF-025.....	50
Tabla 45: Requisito funcional RF-026.....	50



Tabla 46: Requisito funcional RF-027 .....	51
Tabla 47: Requisito funcional RF-028 .....	51
Tabla 48: Requisito funcional RF-029 .....	51
Tabla 49: Requisito no funcional RNF-001 .....	51
Tabla 50: Requisito no funcional RNF-002 .....	51
Tabla 51: Requisito no funcional RNF-003 .....	52
Tabla 52: Requisito no funcional RNF-004 .....	52
Tabla 53: Requisito no funcional RNF-005 .....	52
Tabla 54: Requisito no funcional RNF-006 .....	52
Tabla 55: Requisito no funcional RNF-007 .....	52
Tabla 56: Requisito no funcional RNF-008 .....	52
Tabla 57: Requisito no funcional RNF-009 .....	53
Tabla 58: Requisito no funcional RNF-010 .....	53
Tabla 59: Requisito no funcional RNF-011 .....	53
Tabla 60: Requisito no funcional RNF-012 .....	53
Tabla 61: Requisito no funcional RNF-013 .....	53
Tabla 62: Requisito no funcional RNF-014 .....	54
Tabla 63: Requisito no funcional RNF-015 .....	54
Tabla 64: Requisito no funcional RNF-016 .....	54
Tabla 65: Requisito no funcional RNF-017 .....	54
Tabla 66: Requisito no funcional RNF-018 .....	54
Tabla 67: Requisito no funcional RNF-019 .....	54
Tabla 68: Requisito no funcional RNF-020 .....	55
Tabla 69: Requisito no funcional RNF-021 .....	55
Tabla 70: Requisito no funcional RNF-022 .....	55
Tabla 71: Plantilla para los casos de uso .....	55
Tabla 72: Caso de uso CU-001 .....	56
Tabla 73: Caso de uso CU-002 .....	57
Tabla 74: Caso de uso CU-003 .....	57
Tabla 75: Caso de uso CU-004 .....	58
Tabla 76: Caso de uso CU-005 .....	58
Tabla 77: Caso de uso CU-006 .....	59
Tabla 78: Caso de uso CU-007 .....	59
Tabla 79: Matriz de trazabilidad del sistema .....	60
Tabla 80: Clase Archivo .....	62
Tabla 81: Clase Peticion .....	67
Tabla 82: Clase Respuesta .....	69
Tabla 83: Vista Inicial .....	70
Tabla 84: Vista Principal .....	71
Tabla 85: Vista Respuesta .....	73
Tabla 86: Clase Inicial .....	74
Tabla 87: Clase VistaPrincipalController .....	75
Tabla 88: Clase VistaRespuestaController .....	78
Tabla 89: Clase ChoiceBoxUtils .....	80
Tabla 90: Clase FechaUtils .....	81
Tabla 91: Clase GuardarUtils .....	82
Tabla 92: Clase ObservableListUtils .....	82

Tabla 93: Clase PeticionUtils .....	83
Tabla 94: Clase RespuestaParser .....	83
Tabla 95: Clase StackedBarChartUtils .....	84
Tabla 96: Clase TableViewUtils .....	84
Tabla 97: Clase Utils .....	85
Tabla 98: Plantilla para los casos de prueba .....	94
Tabla 99: Caso de prueba CP-001 .....	95
Tabla 100: Caso de prueba CP-002 .....	96
Tabla 101: Caso de prueba CP-003 .....	96
Tabla 102: Caso de prueba CP-004 .....	97
Tabla 103: Caso de prueba CP-005 .....	98
Tabla 104: Caso de prueba CP-006 .....	98
Tabla 105: Caso de prueba CP-007 .....	99
Tabla 106: Caso de prueba CP-008 .....	99
Tabla 107: Caso de prueba CP-009 .....	100
Tabla 108: Caso de prueba CP-010 .....	100
Tabla 109: Matriz de trazabilidad de los casos de uso y los casos de prueba .....	100
Tabla 110: Resultados de la verificación del sistema .....	101
Tabla 111: Versiones del documento .....	113
Tabla 112: Tareas de la última iteración .....	114

## Glosario de términos

En esta sección se recopilan y describen los acrónimos y los términos usados en la memoria a modo de breve descripción aclaratoria.

<b>Acrónimo</b>	<b>Descripción</b>
<b>AWT</b>	Abstract Window Toolkit, primera librería de interfaces de Java
<b>IDE</b>	Integrated Development Environment o entorno integrado de desarrollo
<b>IVA</b>	Impuesto de Valor Añadido
<b>JVM</b>	Java Virtual Machine, máquina de java para poder ejecutar programas
<b>MVC</b>	Modelo Vista Controlador
<b>OCR</b>	Optical Character Recognition o reconocimiento óptico de caracteres
<b>URL</b>	Uniform Resource Locator, comúnmente llamado dirección de internet
<b>RAE</b>	Real Academia Española
<b>SUS</b>	System Usability Scale o escala de usabilidad del sistema
<b>XML</b>	eXtensible Markup Language

Tabla 1: Acrónimos

<b>Término</b>	<b>Descripción</b>
<b>Adobe Acrobat</b>	Programa de edición y lectura de archivos PDF o Portable Document Format desarrollado por Adobe
<b>Apache Ant</b>	Software de automatización usado durante las fases de compilación y build, desarrollado en Java y que usa sintaxis XML
<b>Bytecode</b>	Código intermedio al que se traduce Java para su ejecución en la máquina virtual
<b>Compilador</b>	Programa traductor cuya misión es traducir de un lenguaje de programación a otro distinto de menor nivel que el anterior
<b>ControlsFX</b>	Librería de programación para crear cuadros de diálogo con JavaFX
<b>Ensamblador</b>	Lenguaje de programación de muy bajo nivel basado en instrucciones entendibles por circuitos integrados, procesadores, etc.
<b>Framework</b>	Estructura tecnológica de soporte con artefactos o módulos software concretos para la organización y desarrollo
<b>Intérprete</b>	Programa capaz de ejecutar otros programas y diferenciado de los compiladores por traducir el programa a medida que es necesario
<b>JavaFX</b>	Librería de interfaces gráficas propiedad de Oracle para el lenguaje de programación Java
<b>jsoup</b>	Librería de programación que ayuda a encapsular archivos HTML en objetos Java mediante
<b>Machine readable data</b>	Datos estructurados de tal forma que sirvan de entrada adecuada a programas informáticos con el objetivo de ser tratados por una máquina
<b>MatLab</b>	MATrix LABoratory, programa desarrollado por Mathworks para procesamiento matemático que cuenta con un entorno de programación y un lenguaje propio llamado M
<b>Microsoft Excel</b>	Aplicación de hojas de cálculo que forma parte de la suite ofimática de Microsoft con objetivos matemáticos y financieros
<b>Parser</b>	Programa o librería que dados unos datos de entrada los estructura de forma que sea más sencillo trabajar con ellos

<b>Scam</b>	Término en inglés usado para hacer referencia a una estafa habitualmente a través de correos electrónicos o páginas web fraudulentas
<b>Screen scraping</b>	Conjunto de técnicas y herramientas de extracción de la información enfocadas a extraer información de programas
<b>Web scraping</b>	Conjunto de técnicas y herramientas de extracción de la información enfocadas a extraer información de páginas web
<b>WebKit</b>	Motor de renderizado para páginas web usado en una amplia variedad de navegadores

Tabla 2: Términos

## 1 Introducción

Nuestra sociedad y nuestra forma de vida exigen de forma constante el uso de la electricidad. Nuestras labores cotidianas, las labores industriales y las empresariales están fuertemente ligadas al consumo energético. Dichas labores involucran distintas acciones y hábitos que son fácilmente predecibles y repetidos por la gran mayoría de las personas en los mismos periodos de tiempo. Es decir, al ir a trabajar o estudiar, todos tenemos unos horarios similares al igual que los horarios de comidas y sueño. Todas estas acciones conllevan un consumo energético asociado. El hecho de que estas acciones sean más o menos rutinarias se traduce en un consumo eléctrico con una curva de la demanda totalmente definida por estos horarios, con momentos en los que el consumo es mínimo (por las noches) denominados valles y momentos en los que el consumo es máximo (al llegar a casa después de estudiar o trabajar) denominados horas punta [CONSUMO]. En la figura 1 se puede observar una curva de la demanda de una jornada laboral, dividida por los distintos sectores significativos de consumo; el residencial, el empresarial y la industria.

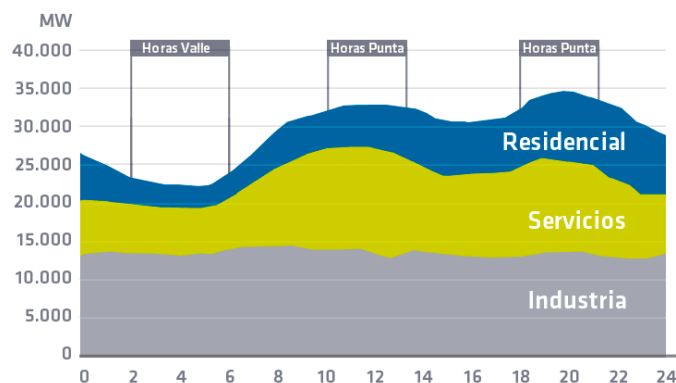


Figura 1: Curva de la demanda de un día normal

La **gestión de la demanda** es la planificación e implementación de medidas destinadas a influir en el modo de consumir energía con el fin de modificar el perfil de consumo, o lo que es lo mismo, una serie de medidas dedicadas a hacer este consumo más eficiente y las curvas del mismo más equilibradas, eliminando valles y horas punta en la medida de lo posible [GESTION].

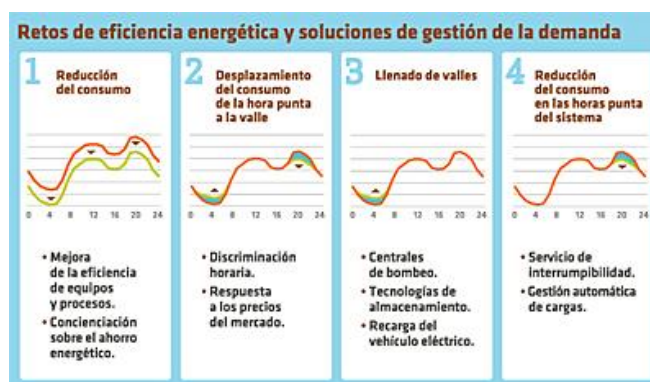


Figura 2: Soluciones de la gestión de la demanda

En la figura 2 se observan las respuestas que aporta la gestión de la demanda para mejorar la eficiencia energética en la actualidad. A la hora de aplicar estas respuestas, es necesario realizar estudios sobre la curva de la demanda real, para lo cual es necesario obtener datos de consumo. Estos datos son publicados por Red Eléctrica de España, a través de su web de transparencia [ESIOS], a fin de que el público interesado pueda consultarlos.

### 1.1 Definición del problema

Como parte de sus labores de investigación, el departamento de **Ingeniería Eléctrica de la Universidad Carlos III** [INGELEC] se enfrenta diariamente a la necesidad de extraer y analizar grandes volúmenes de datos sobre el rendimiento y gestión de la demanda de energía eléctrica a fin de proponer una gestión más eficiente y sostenible del sistema eléctrico. Los datos necesarios como se exponía anteriormente, se ponen a su disposición a través de la web de transparencia de Red Eléctrica Española, sin embargo, aunque disponibles, estos datos se encuentran desestructurados y desagregados, lo cual dificulta su extracción y posterior análisis en herramientas como MatLab [MATLAB]. Por ejemplo, ante la necesidad de analizar los datos sobre la demanda para todo un año se ven en la obligación de obtener todos y cada uno de los informes que lo conforman de manera manual tal y como se puede ver a modo de ejemplo en la figura 3. Esto puede ser sencillo en caso de necesitar los datos en forma de mes, pues sólo sería necesario obtener un informe, pero en el caso de necesitar los datos de forma horaria, se deberían obtener tantos informes como días tiene el año, lo que resulta demasiado laborioso e ineficiente.

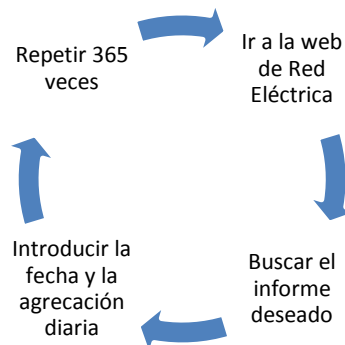


Figura 3: Proceso de obtención de los datos para todo un año

Adicionalmente al problema de la extracción, antes de poder usar los datos extraídos para sus investigaciones, deben realizar un tratamiento sobre los mismos con el fin de usarlos como entrada para programas como MatLab.

### 1.2 Objetivos

El objetivo principal del presente trabajo de fin de grado reside en el **desarrollo de una herramienta software para la extracción de datos sobre el rendimiento de la red eléctrica**. Este objetivo principal se puede desglosar a continuación en tres sub-objetivos principales:

- **Agregación:** el programa permitirá la extracción de los datos existentes en la web de Red Eléctrica de forma agregada y adaptada a las necesidades del departamento.
- **Visualización:** tras la extracción, el programa permitirá la visualización de los datos tanto de forma numérica como de forma gráfica con el objetivo de su análisis.

- **Exportación:** los datos extraídos se podrán exportar en distinto tipo de formatos con el objetivo de no tener que realizar ningún tratamiento posterior de los mismos.

### 1.3 Fases del desarrollo del proyecto

---

El desarrollo del proyecto está basado en la aplicación de un ciclo de vida tradicional con retroalimentación, lo que coloca el esquema organizativo a medio camino entre las metodologías tradicionales y las metodologías ágiles, con el fin de llegar a un producto final con la participación del cliente durante el desarrollo. Este proyecto responde a la necesidad de un cliente real, el departamento de Ingeniería Eléctrica, por lo que las distintas iteraciones del mismo han estado supervisadas por dicho cliente. Las fases que ha atravesado el proyecto son las siguientes:

- **Estudio previo:** en esta fase se realizan entrevistas con el cliente de cara a establecer sus necesidades básicas con el objetivo de solucionar el problema que se ha propuesto.
- **Estudio de la situación:** corresponde al estado del arte. En esta fase se analizan soluciones que por sus características sean similares o usen las mismas tecnologías además de la posible competencia si es que existe, además de las tecnologías disponibles de cara al desarrollo de la herramienta.
- **Análisis:** en el análisis se realiza la toma de requisitos del sistema junto al cliente para definir de una forma más aclaratoria las características clave con las que debe contar la herramienta para la correcta realización de la fase de desarrollo, además de las limitaciones que debe tener el mismo.
- **Diseño:** en esta fase no participa el cliente. Es la fase en la que se establece como se va a realizar la herramienta, habiendo definido la situación actual y las necesidades. Se trata de la fase clave para guiar la implementación de la herramienta, donde se definen de forma detallada todos los componentes del sistema, desde la arquitectura hasta su interfaz.
- **Implementación:** la fase de implementación sigue los puntos establecidos en las fases anteriores del desarrollo de cara a programar la herramienta correctamente. El objetivo de esta fase es cumplir los requisitos definidos en la fase de análisis siguiendo los esquemas y las guías que se han propuesto en la fase de diseño.
- **Evaluación:** es la última fase del proyecto. Aquí se pretende demostrar que la implementación llevada a cabo siguiendo las guías propuestas con anterioridad satisface las necesidades del cliente y cumple los requisitos establecidos de manera correcta.

### 1.4 Medios empleados

---

Para el desarrollo de la herramienta se ha usado tanto artefactos software como hardware. El hardware utilizado ha sido un ordenador de sobremesa con Windows 8.1 corriendo sobre un Intel Core 2 Quad con 8GB de memoria RAM DDR2. En cuanto al software usado se ha centralizado en herramientas pertenecientes a Oracle al ser una herramienta Java, siendo estas herramientas el Java Development Kit en su versión 8 y JavaFX Scene Builder para la creación de la interfaz de usuario.

Toda la herramienta ha sido programada con el entorno de desarrollo Eclipse en su versión Luna. Para la evaluación de la aplicación final se ha usado un total de diez ordenadores ejecutando distintos sistemas operativos.

La presente memoria ha sido redactada usando el editor de texto Microsoft Word 2013 con la ayuda de Adobe Fireworks CS6 para ciertos elementos gráficos.

## 1.5 Estructura de la memoria

---

El presente documento se encuentra estructurado en seis capítulos y seis anexos. A continuación, se describe brevemente el propósito de cada uno de ellos.

En el **primer capítulo** se introduce al lector en la problemática existente, la necesidad de llevar a cabo una solución y los pasos que se han dado para solucionarlo, es en definitiva el apartado de introducción en el cual se escriben estas palabras.

El **segundo capítulo** se trata del estado de la cuestión. En este capítulo se hace un estudio completo del contexto que rodea al mundo de la obtención de datos desde las técnicas antiguas y tradicionales hasta las técnicas informáticas usadas en la actualidad, haciendo mención a algunas herramientas conocidas en este sector. Adicionalmente, se hará una completa revisión de las tecnologías existentes para tomar una decisión razonada sobre la implementación de la solución futura.

En el **tercer capítulo** se presenta la gestión del proyecto, punto en el cual se realiza un análisis del proyecto desde el punto de vista organizativo, estimando las tareas y los recursos de todo tipo necesarios. En este punto también se realiza un análisis de los costes del proyecto en forma de presupuesto.

El **cuarto capítulo** es el más relevante, pues en él se expone de principio a fin la realización de la herramienta generada con el fin de solucionar el problema existente. En un primer apartado, se realiza una definición de la herramienta para poner en contexto al lector y a continuación se define el modelo de programación usado y cada una de las fases que lo componen en detalle, siendo estas el análisis, el diseño y la implementación. En la fase de análisis se explica la toma de los distintos tipos de requisitos pactada con el cliente así como los casos de uso que definen a la aplicación mientras que en la fase de diseño se definen los esquemas que harán posible su implementación. Teniendo las guías de estas dos fases, en la fase de implementación se hará uso de ciertas tecnologías, las cuales son explicadas junto al proceso en sí.

El **quinto capítulo** es el encargado de verificar que el proceso del desarrollo se ha llevado a cabo de forma satisfactoria. Esto se hará mediante casos de prueba con los que testar la aplicación y un formulario de usabilidad con el que se encuestará a un grupo de usuarios.

El **sexto capítulo** recoge las conclusiones finales del trabajo realizado, en las que se resumirán las aportaciones propias así como las posibles mejoras y la problemática encontrada junto a una apreciación personal.



## 2 El estado de la cuestión

---

El objetivo de este apartado reside en el estudio del contexto que rodea el problema abordado por este trabajo de fin de grado. Con este propósito, en primer lugar, se revisa el concepto de “data scraping” o “extracción de datos”. Una vez descrito se realizará un estudio sobre soluciones similares de “data scraping” y las técnicas más aplicadas, que nos permita establecer las características básicas de la solución propuesta. Finalmente, se realizará una revisión de tecnologías con el objetivo de establecer las seleccionadas para el desarrollo de dicha solución.

### 2.1 La extracción de datos o “data scraping”

---

Según el Oxford Dictionary, la palabra “scrape” se traduce al español como “rascar” o “raspar” [OXDICT], por lo que el concepto de data scraping se podría traducir como rascado o raspado de información o datos a través del uso de diferentes técnicas. Estas técnicas de adquisición de información, en relación a la computación, pueden englobarse bajo el concepto de “data scraper”, el cual hace referencia al conjunto de herramientas informáticas aplicadas de distintas formas, con el fin de obtener información concreta, legible por el ser humano (“human-readable data”), a partir de un conjunto de datos en bruto (muy a menudo almacenados en la web). En particular, estas herramientas pueden ser de muy diversas tipologías como por ejemplo herramientas web, extensiones de otros programas, librerías de programación, etc. [DTASCR]. De esta manera, cabe destacar, que la principal diferenciación entre un “data scraper” y un simple “parser” reside en que la salida obtenida por el scraper debe ser también entendida por un ser humano, en lugar de corresponderse únicamente con la entrada necesaria para otro programa.

Si los data scrapers han aumentado de popularidad sobre las técnicas tradicionales de recopilación de la información (como por ejemplo un simple resumen de texto), ha sido por sus muchas ventajas y sobre todo por la mayor de todas ellas: el ahorro sustancial de tiempo. Si bien no es la única ventaja es, con diferencia, la más importante. Un programa de extracción de información o scraper automatizado y especializado en un campo, puede realizar una tarea de obtención de información en una fracción del tiempo que lo haría una persona. Así mismo, un scraper puede realizar una tarea repetitiva de forma más eficiente que un usuario.

A pesar de todas sus ventajas, los data scrapers acogen cierta polémica sobre su legalidad. Debido a la capacidad de estos programas para extraer información de internet, no sólo fragmentos sino incluso webs al completo, suele ser discutido si es lícito o no su uso, ya que la información extraída a menudo contiene derechos de autor que está en mano del usuario respetar. De esta forma, nace una polémica donde las herramientas que se ponen a disposición del público pueden ser usadas para cometer actos ilegales a pesar de que su planteamiento inicial no fuera este. En otras palabras, la legalidad de la extracción de la información y su posterior uso, depende de la forma en la que la use el usuario y no de la herramienta en sí, como en tantos otros ámbitos. Un ejemplo de esto se produce muy a menudo en relación al robo de información y su posterior plagio usando scrapers para duplicar páginas web (llamados scraper sites [SCRSTE]). El propósito de esto es ganar tráfico y monetizarlo mediante publicidad o incluso engañar al usuario para estafarle de una u otra forma mediante “scam”.

Al igual que existe la polémica por el robo de datos, también existe una doble moral, ya que grandes compañías como Google, analizan toda la red almacenando y categorizando información, no sólo con el fin de indexar páginas web sino también con la intención de usarlo en beneficio propio para mejorar sus algoritmos. Por ejemplo, el algoritmo de búsqueda de imágenes mediante redes neuronales o el “knowledge graph”, que permite un primer acercamiento a búsquedas semánticas en el famoso buscador y es la base del funcionamiento del famoso servicio de asistencia “Google

*Now*” [KNWGRP]. Al ser este un tema que interesa a los creadores de contenido (si no apareces en Google no existes) no se le critica de igual manera que a un particular que necesite realizar una investigación y se valga de un data scraper para ayudarse por poner un ejemplo.

Habiendo definido el concepto de extracción de datos y todo lo que lo rodea, es el momento de hablar sobre los tipos de data scrapers existentes. En concreto, los data scrapers se van a categorizar teniendo en cuenta la fuente de donde proviene la información extraída. De esta manera, es posible categorizarlos en “screen scrapers” y “web scrapers”.

## Screen Scraping

Los screen scrapers son el primero de los dos tipos de scrapers existentes. Concretamente son programas que hacen uso de las técnicas de extracción de la información para obtener la información relevante a partir de la interfaz de otra aplicación [ENDDBBB].

Originariamente, los screen scrapers se usaban para leer la salida por consola que generaba un ordenador desde otro conectándolos entre sí. En la actualidad, tienen una gran variedad de aplicaciones, siendo dos de ellas muy populares. Los programas reconocedores de texto basan su funcionamiento en un screen scraper, ya que desde una imagen (obtenida mediante una fotografía, un escaneo o una captura de pantalla), extraen el texto que reconocen y lo convierten a caracteres. Los programas para compartir pantalla y uso remoto también beben de estas técnicas duplicando todo lo que un dispositivo produce y reproduciéndolo en otro. Algunos ejemplos de screen scrapers son los siguientes:

### Sikuli

Este programa, usa las técnicas de screen scraper para automatizar procesos mediante imágenes. Si el usuario no sabe cómo se llama algo de lo que ve en su pantalla, puede introducirlo al programa en forma de captura de imagen para que este haga el procedimiento que quiere. Por poner un ejemplo, para vaciar la papelera del escritorio de nuestro sistema operativo, basta con introducir la secuencia de imágenes capturadas de la acción de vaciar la papelera para que este programa la haga. La figura 4 muestra las indicaciones que necesita Sikuli para buscar y abrir un proyecto de programación en el IDE Eclipse [SIKULI].

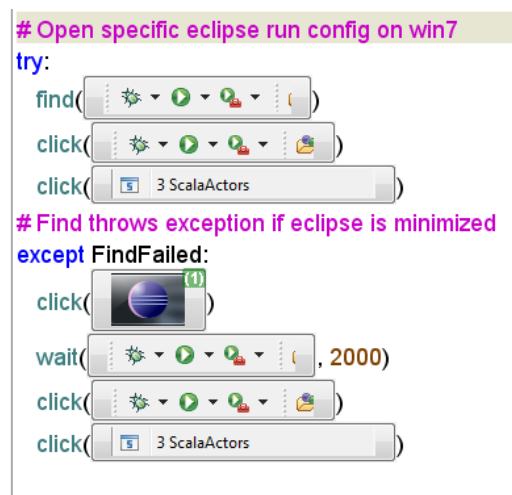


Figura 4: Ejemplo de uso de Sikuli

### *Adobe Acrobat*

Efectivamente, el programa más famoso de Adobe también tiene características de screen scraper. Concretamente el componente que le hace poder considerarse un ejemplo de este tipo de herramienta de recolección de datos es el software OCR (Optical Character Recognition) incluido dentro del programa. El software OCR en general utiliza imágenes como datos de entrada (imágenes escaneadas, imágenes tomadas en el momento mediante una cámara, imágenes almacenadas en memoria, la pantalla de un dispositivo, etc.) para traducirlo más tarde a texto [ADOOCR].

### *TeamViewer*

Este es uno de los programas de acceso remoto más famosos en la actualidad. Los programas de este tipo se caracterizan por copiar el contenido de la pantalla de un dispositivo para trasladarlo a otro y mostrarlo, permitiendo así un uso remoto intuitivo minimizando las diferencias con el uso normal del mismo [TEAM]. Se pueden considerar screen scrapers pues cogen la información que sale por pantalla de un dispositivo para transmitirla al usuario en otro dispositivo y que mediante interfaces de entrada estándares (ratón y teclado o pantallas táctiles) pueda usar el primer dispositivo como si se tratase del segundo. La figura 5 muestra una ventana de TeamViewer en medio de una sesión entre dos usuarios.

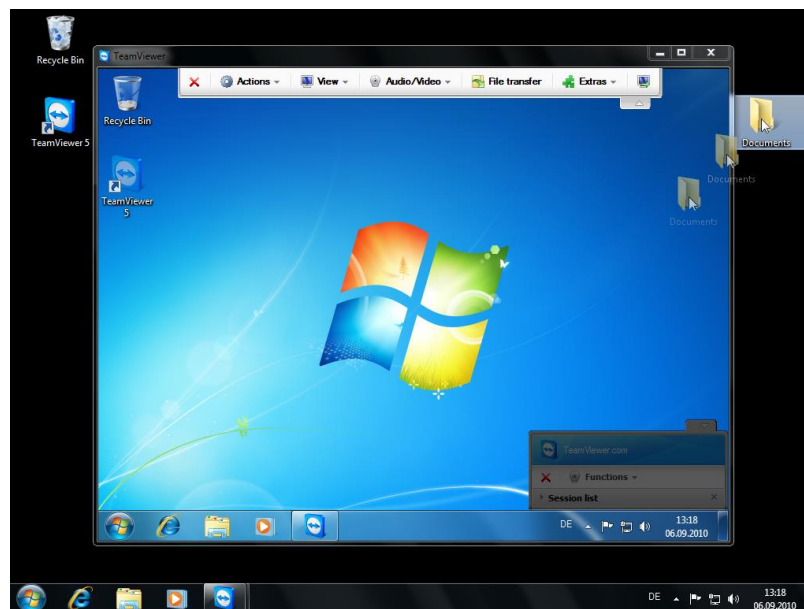


Figura 5: TeamViewer en plena conexión.

### *Web Scraping*

Los web scrapers son el segundo tipo de estas herramientas de recolección de datos. A diferencia de los screen scrapers, los web scrapers se usan para obtener los datos de Internet. Este tipo de scrapers, extraen los datos de forma estructurada de páginas web automática y repetidamente, para más tarde almacenarlos de forma habitualmente estructurada (en un archivo, en una base de datos, como entrada para otro programa, etc.) y preparada para su fácil lectura ya

sea por el usuario o por otro programa. A continuación, se exponen distintos tipos de técnicas que usan los web scrapers para la obtención de la información:

### *Copiar y pegar*

Para obtener la información, algunos web scrapers necesitan la ayuda de un humano que les introduzca los datos que deben ser analizados. Esta técnica si bien no es la más óptima, a veces es la única posible, pues algunas webs implementan barreras para impedir que los scrapers hagan su trabajo evitando así un posible robo de información, como por ejemplo el uso de autenticación en una zona segura donde se encuentre la información reservada a algunos usuarios.

### *Expresiones regulares*

Otra de las técnicas existentes es la obtención del texto plano de una web y su filtrado mediante expresiones regulares, lo que consigue encontrar resultados similares o que cumplan una estructura concreta.

### *Peticiones HTTP*

A veces para obtener datos de la web, es necesario usar técnicas que no podría usar un humano. Las peticiones HTTP a menudo ayudan a encontrar datos que no están disponibles a simple vista para un usuario. Los scrapers que aplican esta técnica realizan la petición HTTP como si se tratase de un navegador web y reciben los datos concretos que han pedido. Como punto negativo cabe destacar que reciben la información legible para una máquina, no para un usuario.

### *Parser HTML*

Cuando los datos que se quieren extraer provienen de una fuente estructurada y reiterativa como un directorio de internet, un parser HTML es la mejor opción. Significa literalmente un analizador de HTML. Los parser HTML están especializados en recibir una estructura concreta de datos para optimizarla, almacenarla, etc. Al contrario que las peticiones HTTP, los parsers devuelven información adaptada a la lectura humana.

### *Tipos de web scrapers*

Los web scrapers pueden clasificarse en varios tipos distintos en función del tipo de entorno de despliegue. Algunos de ellos no son necesariamente aplicaciones web sino librerías para crear otras aplicaciones. Debido a que el número y variedad de web scrapers es muy alto, en esta sección se ejemplifican los más populares y relevantes de cada uno de los tipos:

### *Aplicaciones nativas*

Las aplicaciones nativas o de escritorio, son las aplicaciones programadas desde cero para un sistema operativo concreto. No son programas que se ejecuten pasando desapercibidos para el usuario sino que más bien se ejecutan por sus propias órdenes. Las aplicaciones usadas día a día en ordenadores personales son de este tipo, como por ejemplo el procesador de textos usado para redacta esta memoria, Microsoft Word. Son la evolución lógica de las aplicaciones ejecutadas en consola de la época en la que las interfaces gráficas aun no existían.

Las aplicaciones nativas aportan robustez ya que están diseñadas para el sistema operativo en el que se ejecutan pero a su vez dependen firmemente del mismo, por lo que la portabilidad no es una opción pudiendo coartar la decisión del sistema operativo que el usuario desea utilizar en caso

de que no existan versiones para los más conocidos. Ofrecen una interfaz intuitiva acorde al sistema en el que se están ejecutando que puede suponer una ventaja en la curva de aprendizaje del mismo.

Dentro de este apartado se debe hacer mención a las más o menos recientes aplicaciones móviles que a pesar de no ejecutarse en un sistema operativo convencional con sistema de escritorio, también son nativas (están programadas para el sistema operativo móvil) y cuentan con interfaz gráfica.

- WebHarvy

WebHarvy es un ejemplo de aplicación nativa que realiza las funciones de web scraper. Desde su interfaz permite navegar por internet mientras obtiene la información que el usuario le marca como relevante. Puede obtener todo tipo de información contenida en internet. En la figura 6 podemos ver al programa extrayendo información de la página web de Páginas Amarillas en inglés.

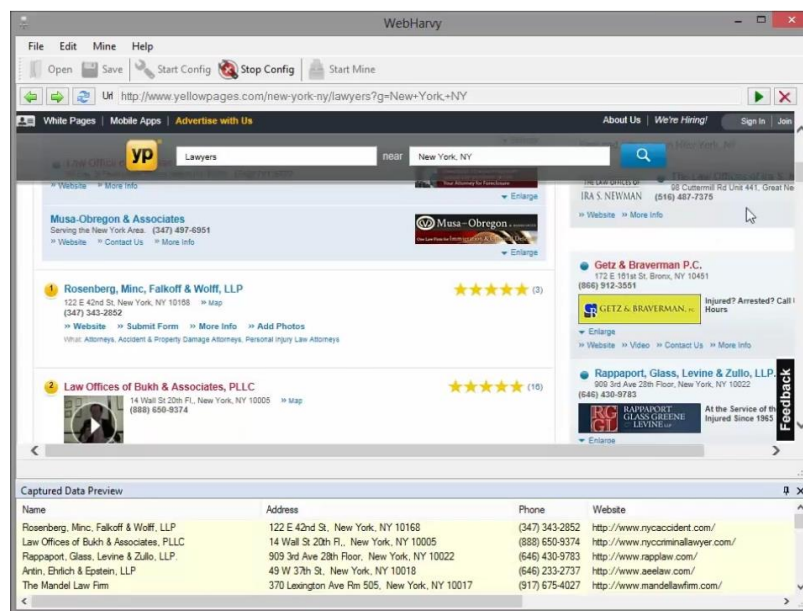


Figura 6: WebHarvy obteniendo información de la web de Páginas Amarillas

- Web Data Extractor

Al igual que el anterior, este programa hace las funciones de web scraper como aplicación nativa. Permite crear reglas con expresiones regulares y buscar en webs visualmente para indicarle al programa lo que necesita el usuario [WEX]. En la figura 7 podemos ver al programa trabajando para extraer información de distintos sitios web.

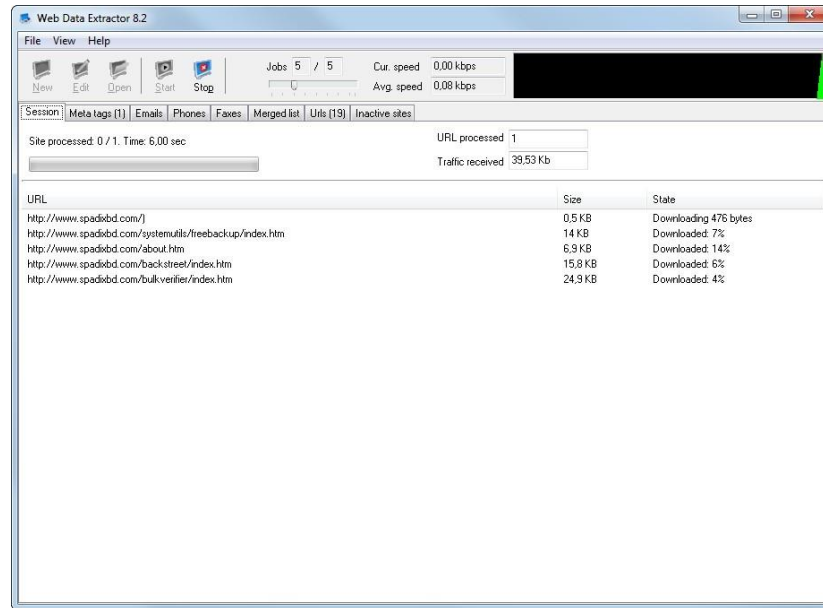


Figura 7: Pantalla principal de Web Data Extractor

### Aplicaciones web

HTML5 es un potente estándar que ha demostrado estar a la altura a la hora de crear aplicaciones de todo tipo, convirtiendo las aplicaciones web en una alternativa real a la hora de plantearse una tecnología de desarrollo.

Sobre sus predecesoras, las aplicaciones nativas o de escritorio, aportan algo muy importante; la portabilidad, al contrario que estas, no está programadas para un sistema operativo concreto. Una aplicación web por su naturaleza, sólo necesita ejecutarse en un navegador web que soporte el estándar en el que se basa (en la actualidad cualquier navegador soporta HTML5 si bien algunos aun no soportan el 100% de sus características) por lo que independientemente del sistema operativo del usuario, la aplicación web podrá ser ejecutada siempre que cuente con un navegador web instalado. En el lado de los contras se encuentra la necesidad de la conexión a internet, ya que la gran mayoría de ellas se ejecutan desde un servidor al que se conecta el navegador.

- Import.io

Se trata de una web que dada una URL proporciona los datos que contiene de forma estructurada [IMIO]. Esta web puede atajar el uso de programas nativos pero no funciona con el 100% de las webs, por poner un ejemplo, para scrapear la web de la UC3M haría falta descargar la aplicación nativa de la misma empresa. En la figura 8 se aprecia la página inicial en la que se describe el funcionamiento y se invita a probarlo con cualquier web.

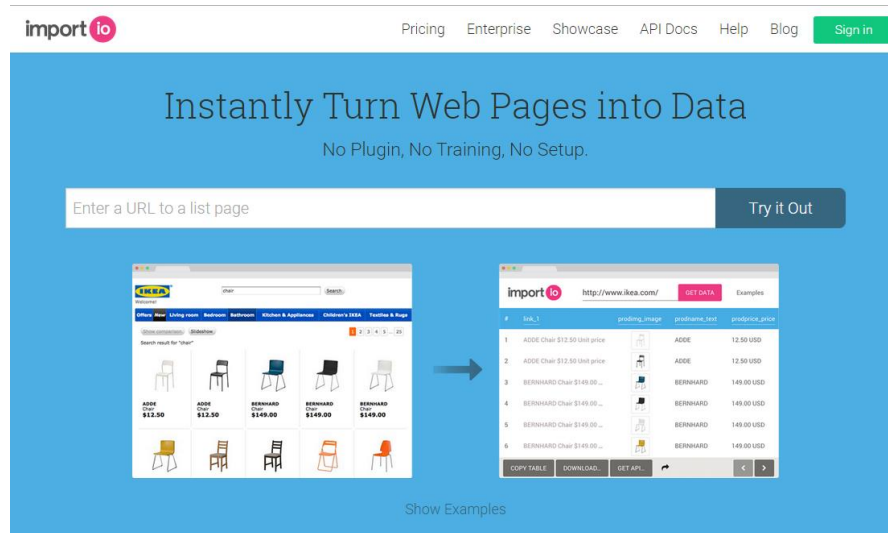


Figura 8: Página principal de import.io

- ScraperWiki

Dentro de esta categoría, ScraperWiki es otra web que permite mediante un navegador recopilar datos. Adicionalmente esta web permite contratar servicios profesionales que van más allá de su versión web [SCRWI].

### *Extensiones de navegador*

La mayoría de los navegadores permiten instalar extensiones, es decir, pequeñas aplicaciones que mejoran el comportamiento o expanden el navegador en sí de alguna forma que puede ser útil para el usuario. Las hay de muchos tipos, pero todas tienen en común las características de las extensiones. La portabilidad en este caso, no se refiere al sistema operativo si no al navegador en sí, es decir, siempre que un navegador cuente con versiones para distintos sistemas operativos, la extensión se podrá instalar en todos ellos, pero para que una extensión sea portable, debe contar con versiones para distintos navegadores o de lo contrario obligaría al usuario a usar un navegador concreto coartando su libertad de elección.

A diferencia de las aplicaciones web, aquí no tiene sentido criticar la conexión a internet necesaria, pues se asume al estar usando un navegador. Si puede comentarse en este caso lo reducido de su utilidad, ya que queda enfocada a algo muy concreto dentro de otra aplicación más grande.

- iMacros

Esta extensión permite automatizar tareas (crear macros) en la web grabando nuestras acciones y extrayendo información en el proceso. Mezcla ambos tipos de data scraper pues por un lado se sirve de la visión del usuario de la web en lugar de la estructura para navegar por la misma y por otro lado extrae información de la web en la que se encuentra [IMAC]. En la figura 9 se puede ver la extensión extrayendo datos desde su propia web en Chrome.



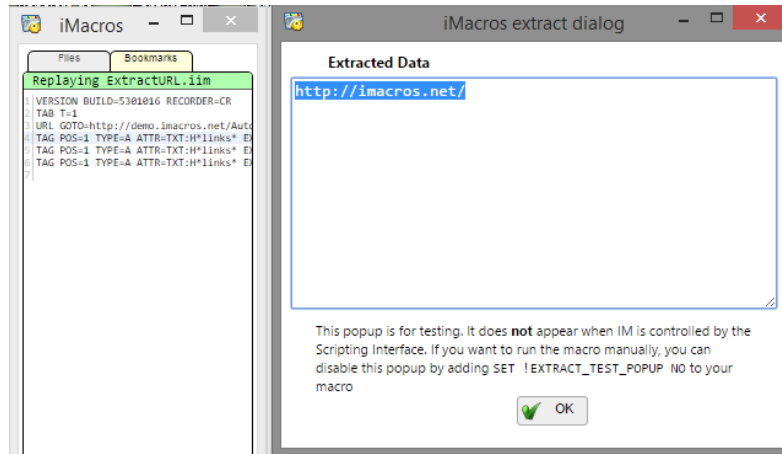


Figura 9: iMacros extrayendo información en Chrome

- Firebug

Firebug comenzó como una extensión del navegador Firefox, que permitía buscar dinámicamente en el código de la web en la que estamos, permitiendo hacer cambios en vivo y revisar la consola, estilos... y todo lo relacionado con la web en la que se ejecuta. Se le considera web scraper porque permite acceder a información de la página web visitada, información que aun estando disponible para el usuario, sería bastante compleja de obtener al completo, de esta forma se pueden hacer cambios de forma sencilla y extraer los datos interesantes. En la figura 10 se muestra a Firebug inspeccionando la web de Red Eléctrica.

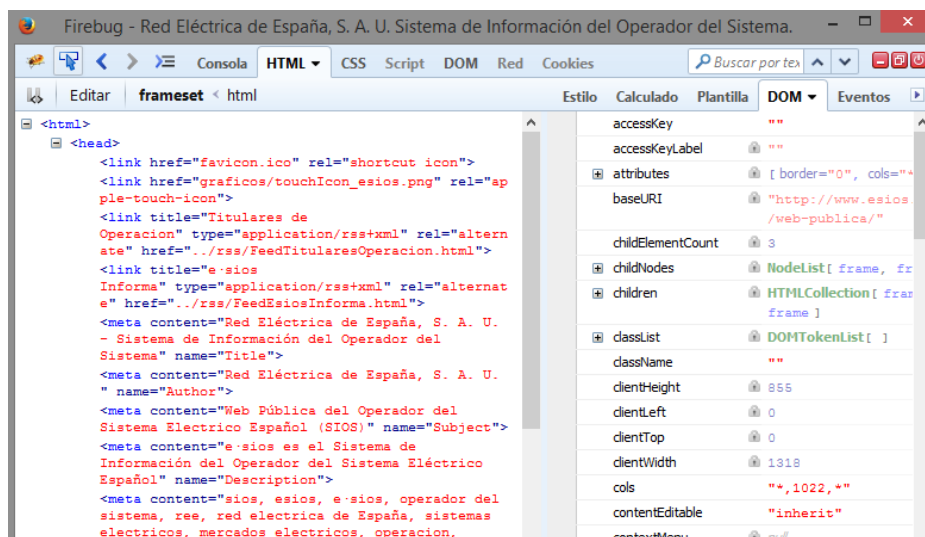


Figura 10: Vista de la web de Red Eléctrica desde Firebug para Firefox

Es el referente en los inspectores de código dentro de los navegadores. Más adelante, los inspectores de código se integraron en el resto de navegadores de forma oficial, como en Chrome y en Internet Explorer por ejemplo, siendo una pieza fundamental para el desarrollo web actual. Por poner un ejemplo de uso, diremos que el inspector de código de Chrome (accesible desde cualquier web pulsando la tecla F12) ha sido usado para la realización de este proyecto, ayudando a encontrar



las peticiones realizadas al servidor desde la web de Red Eléctrica para poder usarlas en el trabajo de forma que sea posible prescindir de la web al completo y convertir la experiencia de uso de la misma en un programa de fácil acceso para el usuario final.

- WebScraper.io

Extensión que se integra dentro de los inspectores de código de los navegadores, aportando una nueva pestaña desde la que se puede extraer fácilmente la información contenida en la web visitada, en lugar de la información visible para el navegador [WSIO]. En la figura 11 se aprecia esta pestaña extrayendo información desde un ejemplo en local.

Sitemaps Sitemap (shop) Create new sitemap				
category_link	category_link-href	title	price	th
Laptops	http://shop.localhost/index.php?id_category=5&controller=category	MacBook Air	504.18	htt ho
Laptops	http://shop.localhost/index.php?id_category=5&controller=category	MacBook	170.57	htt ho
Accessories	http://shop.localhost/index.php?id_category=4&controller=category	Belkin Leather Folio for iPod...	25.04	htt ho
Accessories	http://shop.localhost/index.php?id_category=4&controller=category	Shure SE210 Sound-Isolating...	124.58	htt ho
iPods	http://shop.localhost/index.php?id_category=3&controller=category	iPod Nano	158.07	htt ho
iPods	http://shop.localhost/index.php?id_category=3&controller=category	iPod shuffle	66.05	htt ho
iPods	http://shop.localhost/index.php?id_category=3&controller=category	iPod touch 2	241.64	htt ho
iPods	http://shop.localhost/index.php?id_category=3&controller=category	iPod Nano 2	158.07	htt

Figura 11: Ejemplo de uso de WebScraper en Chrome

### Librerías de programación

Las librerías son una forma útil de añadir características a un programa. Estas herramientas no son programas en sí aunque forman parte de aplicaciones usadas día a día. Ayudan a facilitar las tareas y sirven de comunicación por ejemplo entre una aplicación y su servicio web. Existen de muchos tipos y su principal uso es evitar la redundancia de código o evitar el tiempo de desarrollo que provocaría programar algo que ya existe en una librería.

- cURL

```
(daniel@barrett:pts/2)
1186 | VI: | —% curl -O http://curl.haxx.se/download/curl-7.21.7.tar.bz2 (ACX, 6:43PM)
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
0 2265k 0 11283 0 0 4043 0 0:09:33 0:00:02 0:09:31 23904
```

Figura 12: Ejemplo de uso de cURL en consola

Esta librería de programación no debe ser confundida con el lenguaje de programación de mismo nombre CURL. Se basa en el envío de información con formato URL a través de múltiples protocolos (HTTP, HTTPS, IMAP, IMAPS, LDAP, LDAPS, POP3, etc.), siendo usada en multitud de aplicaciones e incluso aparatos electrónicos como routers, impresoras, equipos de música e incluso coches. En la figura 12 se puede ver la librería en uso desde una consola en Linux.

- Apache Camel

Apache Camel es una librería de programación basada en Java que facilita la mediación entre distintos tipos de lenguajes de programación para la web como el propio Java, Spring, XML, etc. De esta forma es una herramienta muy útil para crear programas y herramientas de web scraping.

- HtmlAgilityPack

Se trata de una librería de programación basada en el framework propietario de Microsoft .NET, que agiliza las tareas de recuperación de la información. No es un programa en sí mismo sino una ayuda para crear programas que necesiten extraer información de internet [HAP]. Ha sido usada durante la carrera en una asignatura.

## 2.2 Elección de las tecnologías de desarrollo

El objetivo de este apartado reside en caracterizar la naturaleza de la solución desarrollada que nos ayude a decidir las tecnologías con las que se van a implementar la solución. La meta del presente trabajo, como ya se ha especificado con anterioridad, es desarrollar una herramienta software que permita extraer y representar datos sobre el rendimiento de la red eléctrica en España, tomando como referencia los datos proporcionados por la página web de Red Eléctrica Española. Teniendo en cuenta esto, lo más adecuado es desarrollar un “web scraper” que permita extraer información de las páginas HTML, la cual luego debe ser posteriormente desplegada al usuario de manera intuitiva. Por lo tanto, las tecnologías seleccionadas tienen que proveer no solo capacidades de extracción de información de páginas HTML, sino también de almacenamiento, interacción y visualización de información.

### Tipo de aplicación

Tal y como ha sido expuesto, los web scrapers pueden presentarse de distintas formas, desde aplicaciones tradicionales nativas, hasta extensiones de navegador pasando por librerías o aplicaciones web. La primera decisión que debe ser tomada, es entonces la naturaleza de la aplicación. Para ello se debe hacer un estudio de las características clave que pueden afectar al desarrollo o al usuario tales como la portabilidad o la conexión a internet, que limitarían fuertemente el uso de las mismas o la libertad de decisión del usuario (el sistema operativo o el navegador deben ser indistintos, es preferible que sea el usuario el que pueda decidir en lugar de tener que adaptarse a las opciones disponibles).

En cuanto a la aplicación programada para resolver la problemática de este proyecto, existían varias posibilidades de las arriba mencionadas. Habiendo decidido usar un web scraper por la naturaleza del problema y habiendo presentado las ventajas y los contras, falta recordar que características concretas debe tener. La aplicación debe poder extraer los datos de la web de red eléctrica, por lo que la conexión a internet es un requisito para poder usarla. La portabilidad es deseable aunque no imprescindible al igual que la robustez. El usuario objetivo de la aplicación no debe ser el avanzado, tiene que estar familiarizado con un entorno normal de trabajo.

A continuación se muestra a modo de comparativa todos los tipos de web scraper que han sido considerados (a excepción de las librerías pues es necesaria una aplicación completa) con sus puntos fuertes y débiles:

Característica	Aplicación de escritorio	Aplicación web	Extensión de navegador
Portabilidad	No es portable entre sistemas operativos	Si pues depende de un navegador web	Si entre sistemas operativos, no entre navegadores

<b>Robustez</b>	Muy robustas, están hechas para el sistema operativo concreto	No especialmente robustas, dependen de la robustez del navegador, la conexión a internet y la optimización	Dependen de la robustez del navegador y su propia optimización
<b>Conexión a internet</b>	No es necesaria a priori a no ser que el problema a solucionar lo requiera	Totalmente necesaria	Totalmente necesaria
<b>Aceptación por parte del usuario</b>	La aceptación es total, a diario se usan aplicaciones de escritorio	La aceptación es parcial, la web puede resultar un inconveniente para el usuario menos avanzado	La aceptación es reducida, sólo los usuarios avanzados suelen usar extensiones de navegador

Tabla 3: Comparativa de tipo de aplicaciones

### *Conclusión sobre el tipo de aplicación usado*

Habiendo presentado las ventajas y desventajas de cada tipo de aplicación, decidir qué tipo es el más conveniente para resolver el problema que presenta este proyecto. A simple vista, se puede decir que tanto una aplicación de escritorio como una aplicación web son convenientes para solucionarlo. Teniendo en cuenta que posiblemente lo más importante sea que el usuario sepa usarla desde el primer momento, la mejor decisión sería la de aplicación de escritorio, siendo este el tipo que finalmente se escogió para realizar el programa.

Parte de la decisión final también se ha tomado teniendo en cuenta que todo lo que rodea al lenguaje HTML5 no es totalmente estándar a día de hoy, además de que el realizar la aplicación web con esta tecnología implicaría más tiempo de aprendizaje, siendo este otro punto para los inconvenientes.

### *Lenguaje de programación*

Una vez decidido que el programa se va a tratar de una aplicación nativa, es necesario hacer un estudio de las diferentes tecnologías que pueden ser usadas para la realización de la misma. A continuación describiremos diferentes lenguajes de programación que podrían ser de ayuda.

#### *C++*

Lenguaje de programación orientado a objetos híbrido, nacido en 1983 y derivado de C (lenguaje que no necesita presentación). A pesar de contar con varias vertiente internas, finalmente se consiguió estandarizar bajo la norma ISO, surgiendo el lenguaje conocido hoy en día. Fue desarrollado por Bjarne Stroustrup y Bell Labs. Durante la Carrera ha sido usado en alguna ocasión en distintas asignaturas. Como la mayoría de los lenguajes de programación orientada a objetos, comparte muchos de los conceptos de estos, como son las clases, los constructores, etc. [C++].

### *C#*

Lenguaje orientado a objetos propiedad de Microsoft y nacido en el año 2000, cuya estructura y funcionamiento recuerda fuertemente a Java. A pesar de no ser el lenguaje más conocido, cuenta con gran aceptación al ser una buena opción para programar para Windows [C#].

### *Python*

Es un lenguaje interpretado puro que data del año 1991. El hecho de que sea interpretado significa que el código escrito en Python nunca se compila sino que el intérprete traduce el código a ensamblador dependiendo de la máquina en la que se ejecuta. Al igual que en otros lenguajes de este tipo, el código se escribe una vez y se ejecuta en distintas máquinas indistintamente gracias a que el intérprete está portado a cada plataforma, lo le confiere portabilidad [PYTH].

### *Java*

Lenguaje del año 1995 que cuenta con una gran aceptación por parte de la comunidad. Es con diferencia, el lenguaje que más se ha usado durante la carrera y más usado a nivel mundial. Se trata de un lenguaje orientado a objetos fuertemente tipado. Es compilado e interpretado, esto significa que el compilador traduce el código fuente a un código intermedio (bytecode) que más tarde el intérprete JVM o Java Virtual Machine (la máquina virtual de Java instalada y adaptada al dispositivo), traduce a código máquina según lo necesita, es decir, no es el dispositivo el que lo ejecuta sino la máquina de java, lo que consigue que la aplicación sea portable **¡Error! No se encuentra el origen de la referencia..**

### *Otros lenguajes de programación*

Aunque no se va a ahondar en ellos, existen otras alternativas que no se han considerado inicialmente como pueden ser Perl, Visual Basic, Objective-C, etc.

### *Lenguaje de programación usado*

Habiendo tenido en cuenta los lenguajes de programación más conocidos y usados, sería conveniente que el lenguaje usado tuviera alguna característica que la naturaleza de las aplicaciones nativas la dejara excluida por defecto, como por ejemplo la portabilidad.

Si bien las aplicaciones nativas no son portables (el hecho de que existan distintas versiones para cada sistema operativo no significa que lo sean), algunos lenguajes permiten la portabilidad introduciendo una máquina virtual que ejecuta el programa y que está programada para cada sistema operativo, lo que significa que una aplicación escrita en uno de estos lenguajes, sólo se programa una vez pero se ejecuta en todos los sistemas que cuentan con la máquina virtual del lenguaje. El máximo exponente en este aspecto es Java, razón por la cual se dice que es un lenguaje interpretado. Otro aspecto a tener en cuenta es el ranking de lenguajes más usados en la actualidad, pues nunca está de más contar con una comunidad activa que pueda ayudar a resolver las dudas que surjan en un desarrollo. En la actualidad el lenguaje más usado es Java tal y como se puede ver en la figura 13 [TIOBE].

Jun 2015	Jun 2014	Change	Programming Language	Ratings	Change
1	2	▲	Java	17.822%	+1.71%
2	1	▼	C	16.788%	+0.60%
3	4	▲	C++	7.756%	+1.33%
4	5	▲	C#	5.056%	+1.11%
5	3	▼	Objective-C	4.339%	-6.60%
6	8	▲	Python	3.999%	+1.29%
7	10	▲	Visual Basic .NET	3.168%	+1.25%
8	7	▼	PHP	2.868%	+0.02%
9	9		JavaScript	2.295%	+0.30%
10	17	▲	Delphi/Object Pascal	1.869%	+1.04%

Figura 13: Los diez lenguajes de programación más usados

A continuación se muestra una tabla comparativa de puntos clave por lenguaje de los arriba explicados:

Característica	C++	C#	Java	Python
<b>Portabilidad</b>	El código se puede compilar para todas las plataformas pero no se es portable	No es oficialmente portable pero se puede aproximar con librerías extraoficiales	Es portable, es un lenguaje que necesita una máquina virtual que existe para todos los sistemas operativos más usados	Es un lenguaje interpretado, se programa una vez y se ejecuta en cualquier máquina que cuente con el intérprete instalado
<b>Soporte de la comunidad</b>	Comunidad algo menos activa	Comunidad algo menos activa	Lenguaje más usado en la actualidad, una comunidad muy activa	Comunidad bastante activa, es un lenguaje que está ganando fuerza
<b>Facilidad de uso y/o aprendizaje</b>	Al tratarse de un lenguaje con varios estándares y diferentes compiladores es algo complejo aprender	De fácil aprendizaje	De fácil aprendizaje	De fácil aprendizaje
<b>Experiencia personal de uso</b>	Experiencia limitada, usado en algunas	Experiencia limitada, usado en algunas	Experiencia alta, siendo el principal lenguaje usado durante la	Ninguna experiencia

	asignaturas de la carrera	asignaturas de la carrera	carrera y en la vida profesional	
--	---------------------------	---------------------------	----------------------------------	--

Tabla 4: Comparativa de lenguajes de programación

Si bien existían muchas alternativas a parte de las que se han analizado brevemente, en este caso ha primado por completo la experiencia personal en el desarrollo en Java. Esto no le resta valor al lenguaje, que aporta muchas de sus características como la portabilidad o la facilidad para crear interfaces como se ve a continuación.

### Interfaz gráfica de la aplicación

Desde el principio, contar con una parte gráfica que permita al usuario decidir los datos que necesita extraer de forma visual, ha sido un requisito indispensable. Habiendo decidido usar Java como lenguaje de programación, hay diferentes alternativas destacables para realizar la interfaz que van a ser analizadas a continuación.

#### AWT

Abstract Window Toolkit, más conocido como AWT, es la primera librería para crear interfaces en Java. Forma parte de “Java Foundation Classes” [JFC] que es un framework para crear aplicaciones Java portables con interfaz gráfica. Si bien tiene los componentes básicos de toda interfaz, flaquea a la hora de intentar crear una aplicación con componentes avanzados, razón por la cual su uso en la actualidad es bastante reducido pero a su vez es una buena opción a la hora de crear aplicaciones sencillas [AWT].

#### Swing

Librería basada en AWT, la evolución de la misma que entra también dentro de JFC. Hasta Java 6, generaba incompatibilidades con AWT que se vieron solucionadas en dicha versión convirtiéndose desde entonces en la herramienta para crear interfaces de usuario predeterminada en Java. Aun siendo mucho más avanzada que AWT, los años pasan factura y se nota en la falta de algunos componentes avanzados como puede ser tablas que permitan reordenarse [SWING]. La figura 14 representa una interfaz sencilla hecha con esta librería.

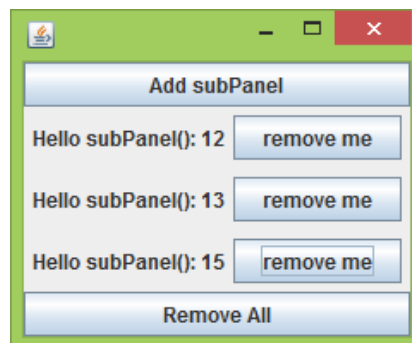


Figura 14: Interfaz sencilla en Swing

#### JavaFX

Esta librería de interfaces es relativamente nueva en comparación con el lenguaje en sí y las otras alternativas presentadas. Se encuentra integrada en Java desde la versión 7. Al contrario que

AWT, cualquier aplicación compilada con Java 7 o posterior, funciona allá donde exista una máquina virtual de Java. Con AWT esto no ocurriría pues para nuevas plataformas, reprogramar AWT es complejo por lo que actualmente no se hace [JFX] .

Se trata de una librería basada en Modelo Vista Controlador, siendo las partes de la interfaz archivos XML con la sintaxis propia de JavaFX y contando con clases atadas a estas vistas que hacen las veces de controlador. Para la parte de la vista, existe un programa oficial que permite construir interfaces de forma visual y con dinámicas de arrastrar y soltar, llamado JavaFX Scene Builder. Al igual que se pueden hacer las interfaces con esta herramienta, se pueden escribir a mano con la sintaxis adecuada. Cuenta con grandes opciones de personalización usando archivos de estilos en cascada (CSS). La figura 15 ilustra la creación de una interfaz de uno de los prototipos de este trabajo con JavaFX Scene Builder.

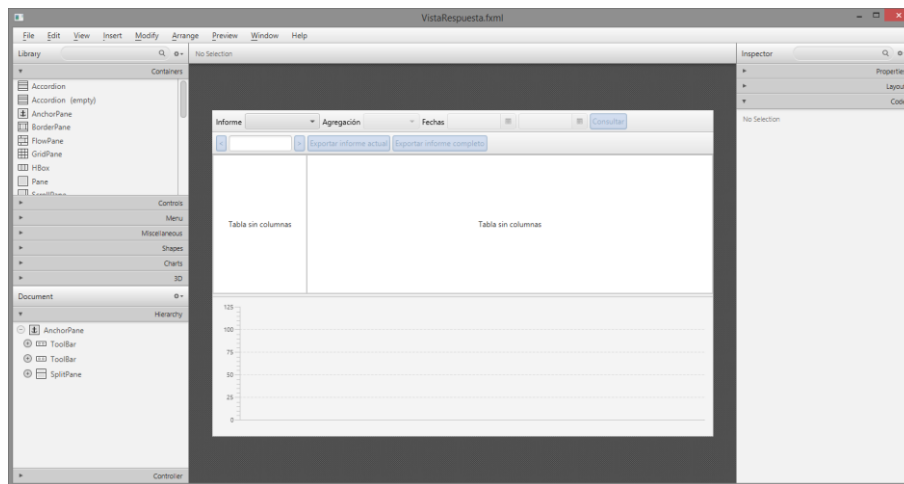


Figura 15: JavaFX Scene Builder con uno de los prototipos de este proyecto

Tiene componentes importantes que se echan en falta en Swing como pueden ser las vistas web integradas usando WebKit. Como heredera de puesto de Swing, es compatible con esta librería, haciendo sencilla la actualización de aplicaciones programadas para ella hacia JavaFX.

### *Librería usada*

Aunque existen algunas alternativas más de las que aquí se reflejan, las más importantes son estas tres. A la hora de hacer una buena interfaz, una parte muy importante es el apartado visual y evidentemente, la usabilidad debe ser un apartado a tener en cuenta. Habiendo examinado las tres opciones, actualmente la que más se aproxima a las necesidades del proyecto es JavaFX, pues permite crear diseños atractivos fácilmente y muy personalizables gracias a JavaFX Scene Builder. A pesar de tener que aprender a programar la parte de los controladores, no es excesivamente complejo y sigue las bases que marcan Java. Otra de las causas de la elección de JavaFX es que las otras dos librerías estén prácticamente obsoletas, teniendo esta ultima una gran cantidad de componentes avanzados que son deseables en las interfaces de hoy en día.

A continuación se muestra una tabla comparativa con varias características de cada librería:

Característica	AWT	Swing	JavaFX
<b>Soporte de la comunidad</b>	El soporte es amplio, es la librería más antigua	Existe bastante soporte, es una librería muy usada	El soporte es medio, no está totalmente aceptada pues acaba de ser integrada en Java
<b>Facilidad de programación</b>	Algo más complejo	Algo más complejo	Muy sencillo con su propia aplicación para crear interfaces de forma visual y estructura XML
<b>Componentes actuales</b>	Faltan por completo	Faltan algunos	Están todos los deseables
<b>Resultado visual</b>	Resultado bastante estándar y mejorable, no cuenta con componentes avanzados	Bastante mejorable, aplicaciones muy parecidas unas a otras y falta de algún componente avanzado	Bastante aceptable y con capacidad de personalización sencilla mediante hojas de estilo

Tabla 5: Comparativa de interfaces para Java

Gran parte del peso de la elección recayó en la novedad de JavaFX frente al resto, al igual que en la facilidad de la construcción de las interfaces mediante la herramienta visual, que si bien no permite abstraerse del todo del diseño sí que lo facilita en gran medida.



### 3 Gestión de proyecto software

En todo tipo de proyecto, la gestión y planificación del mismo es una parte esencial pues permite hacerse una idea de una forma más realista de la magnitud del mismo. Es una forma de establecer los recursos necesarios de tal manera que se asegure la viabilidad del proyecto, para saber de esta forma si es o no rentable y si es factible en cuanto al volumen de tiempo y recursos de los que se pueden disponer. En caso de no cumplir las expectativas, la gestión del proyecto permite decidir si ampliar la inversión el cualquiera de los parámetros (tiempo, presupuesto, recursos humanos, etc.) o abandonarlo al no ser viable.

En esta sección se presenta la gestión de proyecto aplicada a un contexto de negocio real, sin tener en cuenta de que se trata de un trabajo de fin de grado enmarcado en el programa de grado de Ingeniería Informática de la Universidad Carlos III de Madrid. Se realizarán las estimaciones necesarias para llevar a cabo un presupuesto rentable y realista teniendo en cuenta todos los recursos necesarios. Asimismo, dada su creciente aplicación en la mayoría de empresas de desarrollo del software actuales con excelentes resultados frente a las metodologías tradicionales se ha optado por realizar esta gestión de proyecto basada en la aplicación de metodologías ágiles.

#### 3.1 Definición del proyecto

El proyecto aquí descrito, consiste en la realización de una aplicación que obtenga información de forma sencilla y rápida sobre los datos de consumo de la Red Eléctrica Española con el objetivo de ahorrar al usuario tiempo y trabajo. Es deseable que la aplicación sea accesible desde distintos sistemas operativos.

Para realizar esta tarea, se programará un web scraper adaptado a la web que proporcionará al usuario la información obtenida desde la web de Red Eléctrica, parseando los datos extraídos mediante peticiones HTTP de la web, para ser mostrados de forma amigable y más tarde guardarlos de forma estructurada.

#### 3.2 Estimación de tareas y recursos

A la hora de realizar el proyecto, las tareas o fases en las que se debe dividir la organización del mismo para una eficiente realización son las siguientes: estudio previo, análisis, diseño, implementación y pruebas. En algunos proyectos se pacta una fase de mantenimiento durante la cual el cliente puede contar con el equipo de desarrollo para futuros cambios o adaptaciones necesarios con el paso del tiempo. Con el objetivo de realizar estas tareas, son necesarios una serie de recursos humanos comunes a la mayoría de los proyectos de software, a saber: jefe de proyecto, analista, programador y tester.

##### Tareas

Cada tarea es responsabilidad de ciertas personas en función de su cargo y campo. En la tabla 6 se relacionan las distintas tareas con sus correspondientes responsables.

Tarea	Responsables
Estudio Previo	Jefe de proyecto
Análisis	Jefe de proyecto Analista
Diseño	Jefe de proyecto Analista

<b>Implementación</b>	Analista Programador
<b>Pruebas</b>	Jefe de proyecto Tester

Tabla 6: Tareas del proyecto y sus responsables

Con el fin de entender con mayor profundidad cada una de las fases del proyecto, se definen a continuación:

### Estudio previo

En esta tarea el objetivo es recopilar información sobre las tecnologías disponibles en el momento de la realización del proyecto. Se deben tener en cuenta únicamente las que se adecuen a las necesidades de este, es decir, si se trata de una aplicación nativa para sistemas operativos convencionales, se deben obviar las tecnologías móviles. En función de los pros y los contras de cada una de ellas y su nivel de adecuación al proyecto, se debe tomar la decisión de la tecnología que finalmente será usada para su realización. Habitualmente esta tarea cuenta con una subtarea en la que se esboza la solución final, como paso previo a la tarea de análisis. En esta fase participa el jefe de proyecto.

### Análisis

En esta etapa se debe realizar la clásica definición y especificación de requisitos asociada a los proyectos de software, es decir, hay que analizar el problema desde cero observando las necesidades del cliente para obtener los elementos y las acciones que finalmente formarán parte del proyecto, con el objetivo de resolver satisfactoriamente el problema que se plantea desde un principio. Participan el jefe de proyecto y el analista.

### Diseño

En esta fase se diseña desde la arquitectura interna de la aplicación (la parte de modelado y la lógica) hasta la interfaz de usuario. Para realizar esta tarea se debe partir de la fase anterior, cumpliendo al pie de la letra o lo más fielmente posible la extracción de requisitos que se ha realizado con anterioridad (siempre es posible realizar algún cambio de la fase de análisis mientras se realiza el diseño, si bien no es lo más deseable). Los responsables de esta tarea son el jefe de proyecto y el analista.

### Implementación

Se trata de la fase más costosa en tiempo del proyecto. Es en la que se realiza el desarrollo de la aplicación de software teniendo en cuenta los requisitos recopilados en la fase de análisis y las exigencias de la fase de diseño. Esta tarea la realiza exclusivamente el programador con cierta guía del analista.

### Pruebas

Esta tarea es necesaria para poder comprobar que se ha resuelto el problema inicial que provocó la realización del proyecto. Se debe comprobar que se cumplen todos los requisitos definidos anteriormente, de manera satisfactoria y que no quedan errores en la implementación que impidan que el usuario utilice el software de manera exitosa. La fase de pruebas se lleva a cabo por parte de jefe de proyecto y el tester.

## Recursos humanos

De igual manera que quedan definidas las fases del proyecto, para lograr una mayor comprensión de las figuras responsables de cada tarea, se definen a continuación los recursos humanos que dan forma al proyecto:

### Jefe de proyecto

Suele tratarse de un Ingeniero Informático con dilatada experiencia en el campo de los proyectos de software y su dirección. Es el encargado de orquestar todo el proyecto, supervisando cada una de las fases y organizando el equipo humano. Su labor va desde la realización del análisis previo hasta la supervisión de la fase de pruebas. Junto al analista y el cliente, debe definir el proyecto desde cero, en las fases de análisis y diseño teniendo en cuenta las conclusiones del estudio previo.

### Analista

Junto al jefe de proyecto y al cliente, realiza el análisis y el diseño de la solución de software, extrayendo los requisitos y plasmándolos en forma de arquitectura interna e interfaz de usuario. Su tarea es definir el proyecto en función de las necesidades y guiar al programador durante su trabajo. Se trata de un Ingeniero Informático con cierta experiencia en el sector del desarrollo de software sin llegar a tener tanta como el jefe de proyecto.

### Programador

A menudo se trata de un estudiante con cierta experiencia en la programación. Su tarea es seguir las guías establecidas en las fases de análisis y diseño para dar forma a la solución final mediante su tiempo de programación. Cuenta con la ayuda del analista durante su tarea en el desarrollo del proyecto.

### Tester

Es importante aunque no imprescindible que se trate de una persona independiente que no haya participado tanto en la toma de requisitos como en el desarrollo de la aplicación, de tal forma que el proyecto sea lo más desconocido posible para él, consiguiendo así que pueda empezar las pruebas a ciegas. Esto es beneficioso pues una persona que se conozca el proyecto por completo puede no encontrar fallos que una persona totalmente ajena a la definición y desarrollo del mismo encontraría sin problemas. Su tarea pues, es comprobar que la solución desarrollada por el programador, satisface las necesidades del cliente definidas en las tareas anteriores, y que no cuenta con fallos que imposibiliten el correcto uso de la herramienta.

A la hora de estimar los tiempos de cada tarea, tradicionalmente se han seguido ciertas metodologías que eventualmente con la aparición de las metodologías ágiles están siendo desplazadas. El gran inconveniente de las metodologías tradicionales es sobre todo la inflexibilidad de las mismas, inconveniente que provoca que cualquier imprevisto en el desarrollo del software produzca retrasos en el resto de etapas del proyecto.

Las metodologías ágiles surgieron en los años noventa como respuesta a los inconvenientes que presentaban las tradicionales. Se basan en el desarrollo iterativo e incremental, en el que el proyecto se define a la par que se desarrolla consiguiendo así gran flexibilidad. De esta forma, si durante las fases de análisis y diseño se han establecido ciertos requisitos o esquemas que durante la fase de desarrollo se demuestran imposibles, ineficientes o simplemente innecesarios, se pueden modificar de tal forma que no afecten al desarrollo en sí de manera sustancial, tal y como ocurriría

con las metodologías tradicionales. Algunos ejemplos de estas metodologías son “Cristal Clear”, “Extreme Programming” o XP, “Agile Unified Process” o AUP y como gran referente “Scrum” siendo la metodología ágil más usada y conocida en la actualidad, a pesar de datar del año 1986.

En la tabla 7 se pueden apreciar las principales ventajas de las citadas metodologías ágiles frente a las metodologías tradicionales [AGIL], donde se puede observar que son prácticamente antagónicas.

Metodologías tradicionales	Metodologías ágiles
Predictivos	Adaptativos
Orientados a procesos	Orientados a personas
Proceso rígido	Proceso flexible
Se concibe como un proyecto	Un proyecto es subdividido en varios proyectos más pequeños
Poca comunicación con el cliente	Comunicación constante con el cliente
Entrega de software al finalizar el desarrollo	Entregas constantes de software
Documentación extensa	Poca documentación

Tabla 7: Ventajas de las metodologías ágiles frente a las tradicionales

Observando las tendencias actuales, se decide que elaborar una planificación completa incluyendo su diagrama de Gantt es algo desfasado y que en la gran mayoría de los casos termina por incumplirse. Habiendo definido las tareas y sus responsables, se estiman entonces la duración en días de cada una de las mismas sin entrar en más detalles de planificación. La tabla 8 recoge estas estimaciones que han sido elaboradas basándose en proyectos de magnitud similar y la experiencia propia:

Tarea	Duración en días
<b>Estudio Previo</b>	15 días
<b>Análisis</b>	25 días
<b>Diseño</b>	40 días
<b>Implementación</b>	70 días
<b>Pruebas</b>	15 días
<b>Total</b>	<b>165 días</b>

Tabla 8: Tareas del proyecto y su duración aproximada

Teniendo en cuenta que un mes tiene de media 22 días laborables y el total de días estimados es de 165, el proyecto tendrá una duración de siete meses y medio. Puesto que se trata de un proyecto de software, los cuales son susceptibles de tener retrasos, se estima que el proyecto tendrá una duración total de **ocho meses**.

### 3.3 Presupuesto

A la hora de hacer un presupuesto, es necesario realizar un desglose de cada uno de los grupos de elementos que incurrirán en gastos, dividiéndolos en personales, materiales y generales. Como se ha comentado, el presupuesto será adaptado a los ocho meses de duración del proyecto.

## Costes de personal

El equipo humano, definido anteriormente, está integrado por un jefe de proyecto, un analista, un programador y un tester. A la hora de calcular los costes derivados de los salarios se asumen los siguientes puntos:

- El jefe de proyecto cuenta con una experiencia en el sector de 20 años.
- El analista tiene una experiencia profesional de 8 años.
- El programador cuenta con una experiencia de 3 años.
- El tester tiene una experiencia en su puesto de 2 años.
- La jornada laboral estándar se fija en ocho horas para hacer los cálculos medios.
- La semana laboral estándar es de cinco días con el mismo objetivo de realizar los cálculos.
- Las horas laborables, teniendo en cuenta los dos puntos anteriores, serán de 176 al mes de media.
- No se realizarán horas extra como tal, es decir, el máximo de horas diarias dedicado por persona al proyecto viene dado por su contrato, no permitiéndose extenderlo diariamente. De ser necesario se ampliaría la duración del proyecto pero no la de la jornada diaria.
- Los salarios tomados de referencia están adecuados a los salarios medios reales en la actualidad y adecuados a la magnitud del proyecto y experiencia concreta de cada uno de los trabajadores [SAL].

En la tabla 9 se muestran las aproximaciones realizadas teniendo en cuenta todo lo anterior:

Profesión	Salario bruto medio mensual	Salario bruto medio por hora
<b>Jefe de proyecto</b>	3.800 €	21,60 €
<b>Analista</b>	2.500 €	14,20 €
<b>Programador</b>	1.250 €	7,10 €
<b>Tester</b>	1.250 €	7,10 €

Tabla 9: Salarios brutos medios por profesión

La dedicación de cada responsable al proyecto será distinta en función de su puesto; el jefe de proyecto trabajará en el mismo, tres horas al día, el analista y el programador una jornada completa de ocho horas al día y el tester media jornada de cinco horas diarias. Teniendo en cuenta los salarios brutos medios por hora, en la tabla 10 se muestra el salario bruto mensual sabiendo las horas por jornada laboral de cada trabajador y los 22 días por mes de media:

Responsable	Salario por hora	Horas por día	Salario bruto mensual
<b>Jefe de proyecto</b>	21,60 €/hora	3 horas/día	1.425,60 €/mes
<b>Analista</b>	14,20 €/hora	8 horas/día	2.499,20 €/mes
<b>Programador</b>	7,10 €/hora	8 horas/día	1.249,60 €/mes
<b>Tester</b>	7,10 €/hora	5 horas/día	781,00 €/mes

Tabla 10: Salario bruto mensual por trabajador

Habiendo calculado los salarios brutos mensuales de cada uno de los trabajadores, es necesario calcular los costes derivados de la cotización a la seguridad social.

Para este proyecto es interesante tener en cuenta dos grupos de cotización concretos [SEGCAT]. El primero corresponde a ingenieros, licenciados y personal de alta dirección no incluido en el artículo 1.3.c) del Estatuto de los Trabajadores y sus bases mínimas y máximas de cotización son 1.056,90 €/mes y 3.606,00 €/mes respectivamente. El segundo corresponde a ingenieros técnicos, peritos y ayudantes titulados siendo sus bases mínimas y máximas de cotización 876,60 €/mes y 3.606,00 €/mes respectivamente. Al primer grupo pertenecerían el jefe de proyecto y el analista mientras que al segundo el programador y el tester. Puesto que estas bases están calculadas para una jornada estándar, habría que calcular la proporción a la jornada de cada uno de los trabajadores en la tabla 11:

Responsable	Horas por día	Cotización mínima	Cotización máxima
<b>Jefe de proyecto</b>	3 horas/día	396,34 €/mes	1.352,25 €/mes
<b>Analista</b>	8 horas/día	1.056,90 €/mes	3.606,00 €/mes
<b>Programador</b>	8 horas/día	876,60 €/mes	3.606,00 €/mes
<b>Tester</b>	5 horas/día	547,88 €/mes	2.253,75 €/mes

Tabla 11: Bases de cotización por trabajador

A excepción del jefe de proyecto, todos los salarios quedan comprendidos en los rangos de cotización. En el caso del jefe, al superar su salario mensual la base máxima de cotización, cotizará sobre dicha base en lugar de su sueldo. El Estado establece que para el año 2015, la empresa debe sufragar un **23,6%** del salario o base de cotización aplicada por cada trabajador [SEGCOT], por lo que los costes de la seguridad social mensual serían los mostrados en la tabla 12:

Responsable	Salario bruto mensual	Coste de la Seguridad Social mensual	Coste total mensual
<b>Jefe de proyecto</b>	1.425,60 €/mes	319,13 €/mes	1.744,73 €/mes
<b>Analista</b>	2.499,20 €/mes	589,81 €/mes	3.089,01 €/mes
<b>Programador</b>	1.249,60 €/mes	294,91 €/mes	1.544,51 €/mes
<b>Tester</b>	781,00 €/mes	184,32 €/mes	965,32 €/mes

Tabla 12: Coste total mensual por trabajador

Para calcular los costes totales de personal habría que hacer la proporción de los gastos mensuales derivados de cada trabajador al tiempo que dedicará al proyecto en total, tal y como se muestra en la tabla 13:

Responsable	Coste total mensual	Tiempo dedicado al proyecto	Coste total
<b>Jefe de proyecto</b>	1.744,73 €/mes	4 meses y 7 días	7.534,06 €
<b>Analista</b>	3.089,01 €/mes	6 meses y 3 días	18.955,29 €
<b>Programador</b>	1.544,51 €/mes	3 meses y 4 días	4.914,35 €
<b>Tester</b>	965,32 €/mes	15 días	658,17 €
		<b>Total</b>	<b>32.061,87 €</b>

Tabla 13: Coste total del personal

En total, los costes de personal ascenderán a **32.061,87 €**.

### Costes de material

Para la realización del proyecto, cada trabajador necesitará un equipo informático completo, tratándose de un portátil Toshiba Satellite L50D-B-14U [TOSAMZ] y un ratón Logitech B100 [LOGAMZ]. Para calcular el coste imputable al proyecto de estos elementos, se debe establecer un periodo de depreciación concreto, que suele estar en torno a los cinco años, es decir 60 meses. Los precios mostrados son sin el 21% de IVA. En la tabla 14 se calculan dichos costes teniendo en cuenta que la duración del proyecto son ocho meses y se compraran cuatro unidades de cada artículo:

Artículo	Coste sin IVA	Coste total	Coste imputable
<b>Toshiba Satellite L50D-B-14U</b>	422,71 €	1.690,84 €	225,45 €
<b>Logitech B100</b>	6,24 €	24,96 €	3,33 €
		<b>Total</b>	<b>228,78 €</b>

Tabla 14: Coste de hardware

En cuanto al software no habrá gastos pues el sistema operativo viene incluido en el precio de los equipos y el resto de software usado será libre. Por otra parte, el proyecto tendrá ciertos gastos asociados en forma de consumibles y bienes fungibles, siendo estos últimos el típico material de oficina como bolígrafos, folios y cuadernos. En la tabla 15 se calculan los gastos totales teniendo en cuenta que la duración del proyecto es de ocho meses:

Consumible	Coste mensual sin IVA	Coste total
<b>Electricidad</b>	55,00 €	440,00 €
<b>Agua</b>	12,00 €	96,00 €
<b>Internet</b>	45,00 €	360,00 €
<b>Materiales fungibles</b>	4,00 €	32,00 €
	<b>Total</b>	<b>928,00 €</b>

Tabla 15: Coste de consumibles

En total, todos los costes materiales ascenderían a los mostrados en la tabla 16:

Material	Coste del material
<b>Hardware</b>	228,78 €
<b>Consumibles</b>	928,00 €
<b>Total</b>	<b>1.152,78 €</b>

Tabla 16: Coste de material total sin IVA

Los costes materiales totales ascenderán a **1.152,78 €** sin incluir el IVA.

### Coste total del proyecto

Habiendo calculado todos los costes, el coste total de personal y material quedaría definido en la tabla 17:

Coste	Importe
<b>Coste de personal</b>	32.061,87 €
<b>Coste de material</b>	1.152,78 €
<b>Total</b>	<b>33.214,65 €</b>

Tabla 17: Coste total bruto del proyecto

A la hora de calcular el coste total del proyecto, se debe tener en cuenta el margen de beneficios deseado y un margen de riesgo asumible por cualquier contingencia que pudiera ocurrir. Para ambos valores se recurre a proyectos similares, decidiendo que los beneficios se situarán en un **15%** y el riesgo en un **10%** (algo normal en proyectos de este tipo). En la tabla 18 se calcula el coste total final del proyecto teniendo en cuenta beneficios, riesgos y el **21%** de IVA establecido por ley:

Coste	Importe
<b>Coste total</b>	33.214,65 €
<b>Beneficio</b>	4.982,20 €
<b>Riesgo</b>	3.321,47 €
<b>Total antes de impuestos</b>	<b>41.518,32 €</b>
<b>IVA</b>	8.718,85 €
<b>Total</b>	<b>50.237,17 €</b>

Tabla 18: Coste final del proyecto

El presupuesto total de este proyecto asciende a la cantidad de Cincuenta Mil Doscientos Treinta y Siete con Diecisiete euros (50.237,17 €) a pagar en un único plazo al comienzo del proyecto.



## 4 Solución

En este apartado, se define la solución llevada a cabo con el fin de resolver el problema expuesto en apartados anteriores. En una primera instancia, se realizará una breve descripción de la solución para, a continuación, realizar una descripción completa del proceso de desarrollo con sus diferentes fases, partiendo del análisis hasta llegar a la implementación.

### 4.1 Descripción de la solución

Como se ha comentado con anterioridad, la solución propuesta en el presente trabajo fin de grado responde a la necesidad planteada por el departamento de Ingeniería Eléctrica de la Universidad Carlos III de Madrid de analizar volúmenes masivos de datos de consumo de la red eléctrica registrados durante diversos períodos temporales. Los datos deseados se pueden obtener de la web oficial de Red Eléctrica ESIOS o “Sistema de Información del Operador del Sistema” [ESIOS], pero a través de este medio la extracción y agregación de grandes volúmenes de datos diarios resulta ineficiente además de faltar un soporte a su análisis y exploración. De esta forma surge la necesidad de resolver el problema mediante otros medios.

Para resolverlo, se ha desarrollado una herramienta software que extraiga los datos de rendimiento de la red eléctrica, permita su agregación de acuerdo a los intervalos temporales seleccionados por el usuario y su posterior visualización de manera adaptada a él y sin apenas intervención por el mismo más que la introducción de los datos necesarios para la consulta, minimizando el trabajo y el tiempo en comparación con la extracción de los mismos datos desde la web oficial. Preferiblemente, la aplicación debe ser portable al menos entre los sistemas operativos de sobremesa mayoritarios.

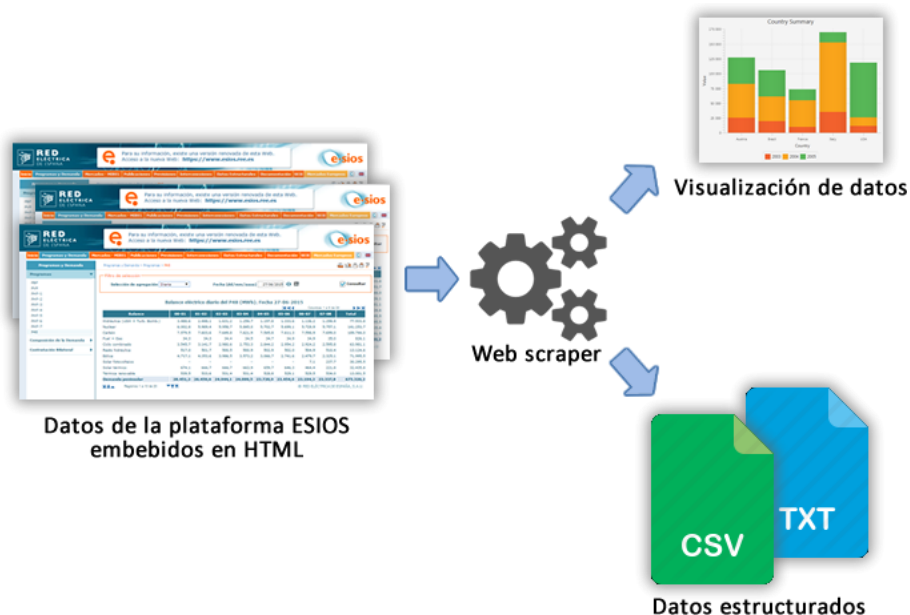


Figura 16: Esquema de la solución

En la figura 16 se muestra de forma gráfica el esquema de la solución que proporciona la aplicación que se ha programado para el problema planteado. El usuario obtendrá los datos deseados realizando el tipo de consulta correspondiente al tipo de informe que desee mediante la

aplicación, que a su vez se encarga de solicitar los datos a Red Eléctrica y transformarlos para la visualización del usuario. Es decir, la aplicación programada se trata de un puente entre el usuario y los datos en bruto de Red Eléctrica facilitando su obtención al usuario.

A continuación, se listan el conjunto de tareas que el usuario podrá realizar sobre la herramienta:

- **Realizar una petición normal:** para realizar un informe, el usuario debe introducir el tipo de informe que desea a elegir de la lista que se le proporciona, la agregación en la que quiere los datos, es decir, si de sea los datos divididos en horas, días, meses o años, y la fecha de la que precisa los datos.
- **Realizar una petición por intervalos:** la gran ventaja de la aplicación reside en la posibilidad de realizar peticiones por intervalos, para ello el usuario debe introducir el tipo de informe que desea, la agregación y las fechas inicial y final del informe. Al usuario se le mostrarán entonces todos los informes contenidos entre ambas fechas.
- **Exportación de informes:** otra ventaja reseñable se trata de la posibilidad de exportar los datos obtenidos, ya sea de un informe sencillo o un informe por intervalos, en diversos formatos que den la posibilidad de tratar los datos en programas de cálculo como MatLab.
- **Previsualización de los datos en forma de tabla dinámica:** nada más realizar el informe los datos se muestran en la tabla de la vista del programa donde se da la opción de reordenar las columnas al gusto del usuario.
- **Visualización de los datos de forma gráfica:** cada uno de los informes lleva asociada una visualización en forma de gráfica de barras apilada en la que se reflejan los datos de consumo obtenidos mediante la petición, de esta forma se pueden analizar los datos numérica y gráficamente de una sola vista.

En la figura 17, se puede observar la interfaz de usuario de la herramienta después de realizar una petición simple.

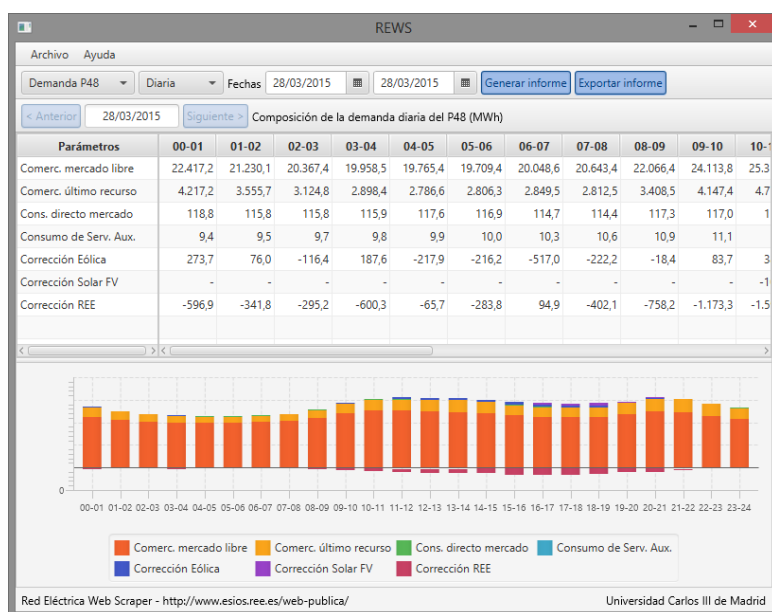


Figura 17: Interfaz de usuario de la aplicación tras realizar una petición simple

La interfaz de usuario de la aplicación cuenta con una perspectiva de bienvenida en la que se muestra un menú superior desplegable con la posibilidad de llegar hasta la pantalla de informes además de un diagrama explicativo del funcionamiento de la aplicación. Una vez en la perspectiva de informes se provee al usuario con un pequeño formulario en la vista superior seguidas de una vista de tabla y una vista de gráfica en la parte inferior. Esta tabla y esta gráfica son las zonas donde se mostrarán los datos de consumo tras rellenar y realizar la petición (el tipo de informe, la agregación y la fecha), tal y como se puede observar en la figura superior. Una vez obtenida la respuesta, es posible exportar el informe para su tratado posterior con otros programas.

Toda la ventana es redimensionable y la proporción entre tabla y gráfica también es personalizable hasta cierto punto, poniendo un límite inferior en cuanto a tamaño para evitar que los datos sean incomprensibles.

Para llegar hasta la versión final, durante todo el desarrollo se ha seguido una metodología que permitía el uso de prototipos, con el fin de obtener una aplicación a gusto del usuario final y que cumpliera todas las características fijadas durante las fases iniciales del desarrollo de forma satisfactoria. Si bien alguna de las versiones intermedias obtenidas cumplía los requisitos del desarrollo, no estaban totalmente a gusto del cliente por lo que se prosiguió el desarrollo hasta la versión final.

Para la correcta ejecución de la aplicación, únicamente hay que abrir el archivo .jar generado al compilar los archivos generados durante del desarrollo. De esta fase se hablará ampliamente en apartados posteriores de este documento.

Un requisito indispensable que ya se ha comentado anteriormente, es la necesidad de tener instalado en el equipo la máquina virtual de Java y disponer de una conexión a internet, como mínimo durante la realización de la petición. Cumpliendo todos estos requisitos, la aplicación será capaz de funcionar con normalidad.

## 4.2 El proceso de desarrollo

---

En este apartado se describe el proceso de desarrollo de la herramienta ideada como solución al problema planteado. Se explicaran todas las fases que ha atravesado el desarrollo del mismo para llegar a la solución final.

### Modelo de proceso

Para el desarrollo de la herramienta, se ha seguido un modelo de proceso tradicional con retroalimentación, lo que puede considerarse como el proceso precursor de las metodologías ágiles, ayudándose de una fase de prototipado donde se han sugerido distintas aproximaciones hasta llegar al resultado final del software. En la figura 18 se puede apreciar el esquema seguido durante el desarrollo, común a los proyectos que siguen este mismo modelo.

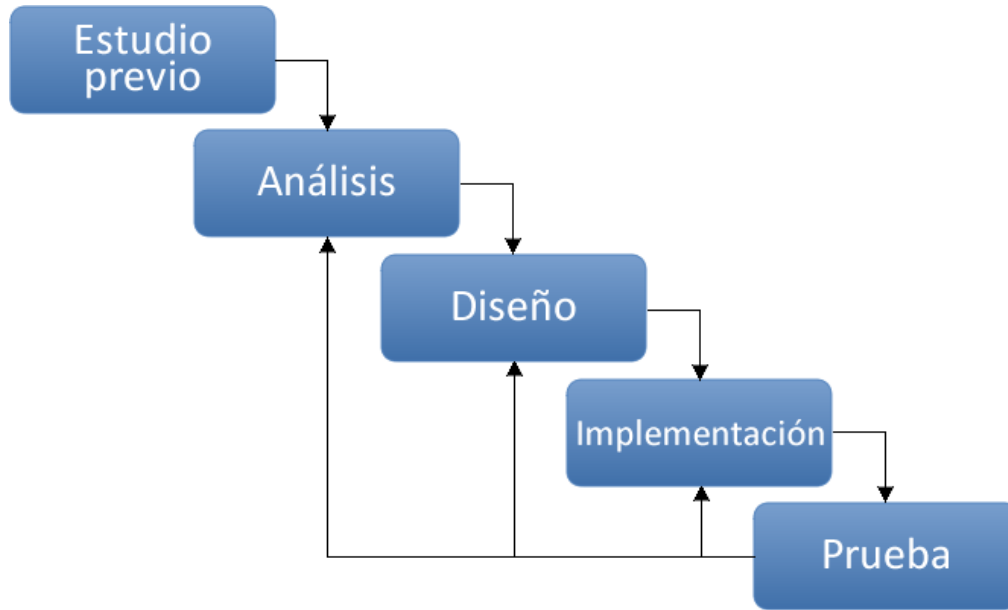


Figura 18: Ciclo de vida tradicional con retroalimentación

El ciclo de vida de un software queda descrito en la norma ISO 12207-1 como el marco de referencia que contiene los procesos, las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto de software, abarcando la vida del sistema desde la definición de los requisitos hasta la finalización de su uso [ISOCICLO]. La finalidad del mismo es garantizar que el software resultante de seguir las fases descritas cumpla los requisitos propuestos durante las primeras fases de desarrollo satisfaciendo las necesidades iniciales del cliente o usuario.

A la vista de la figura anterior, las fases principales del desarrollo son el estudio previo, donde se realiza un análisis de la situación actual y el entorno para ver la viabilidad y necesidad del proyecto con el fin de definir los objetivos del mismo, el análisis propiamente dicho, donde se define la herramienta y los requisitos junto al cliente o usuario objetivo, el diseño, que abarca tanto el diseño de la arquitectura y la lógica interna de la aplicación como el diseño de la interfaz, la implementación, fase en la que se programa la herramienta usando las guías proporcionadas por las fases de desarrollo anteriores y las pruebas, con el fin de comprobar que se han cumplido los requisitos establecidos durante la fase de análisis.

Durante todas las fases y siguiendo el modelo de proceso establecido, es posible volver a fases de desarrollo anteriores para hacer modificaciones concretas que puedan surgir durante las fases posteriores, consiguiendo de esta forma un modelo menos rígido y que admita cambios. Esto permite en todo momento subsanar posibles errores o lacras en el desarrollo que a primera vista en las primeras fases se hubieran pasado por alto. Los desarrollos de software en la actualidad son procesos vivos que necesitan de una flexibilidad más o menos elevada, para poder realizar estos cambios y corregir el rumbo ante un escenario cambiante.

## Análisis

Durante la fase de estudio previo y la propia fase de análisis, se han realizado entrevistas al cliente (el departamento de Ingeniería Eléctrica de la Universidad Carlos III) con el fin de aclarar las necesidades básicas y los puntos deseables a tener en cuenta a la hora del desarrollo de la aplicación. En los apartados siguientes se procede a realizar la toma y especificación de requisitos,

proceso que se llevó a cabo con la ayuda del cliente de la aplicación. En un anexo adjunto a este documento se encuentran las entrevistas realizadas al cliente [Anexo III. Entrevistas con el cliente

Adicionalmente, como en todo sistema existen casos de uso concretos que surgen de los requisitos.

Los tipos de requisitos existentes, los requisitos en sí, los casos de uso y la matriz de trazabilidad del sistema, se explican y muestran a continuación.

### *Definición de requisitos*

La definición de requisitos es una de las partes más importantes del desarrollo, ya que es el momento en el que se recogen las partes necesarias y las partes deseables para el usuario final o el cliente que lo componen.

La importancia de los requisitos se debe a que son la guía a seguir durante las siguientes fases del desarrollo con el objetivo de realizar una herramienta que se adopte a las necesidades y peticiones del cliente. Sin esta guía concreta el desarrollo iría a la deriva.

Tradicionalmente los requisitos se dividen en dos apartados concretos; los requisitos funcionales y los requisitos no funcionales. Los requisitos funcionales son los referentes a las acciones fundamentales que debe realizar la solución de forma exitosa, es decir, lo que permite el software y como lo permite. Por otra parte, los requisitos no funcionales son los referidos a partes más intangibles o que simplemente no representen partes funcionales del desarrollo como por ejemplo seguridad, fiabilidad, portabilidad, etc. Adicionalmente, en algunos casos se pueden definir requisitos referentes a la interfaz en un apartado diferente, pero en este caso lo incluiremos en los requisitos no funcionales.

Para una correcta definición de requisitos, es necesario definir con antelación una plantilla en la que todos puedan estar englobados y cuyo contenido sea claro y conciso para la correcta realización del resto de fases de desarrollo. Para ellos se hará uso de una plantilla estandarizada adaptada a las necesidades propias del proyecto [VOLERE]. La tabla 19 refleja la plantilla que se usará para los requisitos, tanto los funcionales como los no funcionales.

ID	
Prioridad	
Necesidad	
Descripción	
Verificación	

Tabla 19: Plantilla para los requisitos funcionales y no funcionales

A continuación se explican cómo deben rellenarse cada uno de los apartados que componen la tabla de definición de requisitos arriba expuesta:

- **ID:** el identificador es una clave única para cada requisito que no puede ser repetida. Será la forma de referirse a los mismos para que no haya confusión. Se forma empezando con las siglas del tipo de requisito (RF o RNF dependiendo de si es funcional o no funcional), seguido de un guion y un código numérico de tres cifras único para cada uno por estricto orden de creación o necesidad. De esta forma, el código del primer requisito no funcional sería RNF-001 y se incrementaría por unidades para cada uno de los requisitos siguientes del mismo tipo.
- **Prioridad:** este campo indica el grado de urgencia a la hora de la implementación de los requisitos en el proyecto, existiendo tres grados distintos; Alta, Media y Baja.

- **Necesidad:** aunque puede parecer redundante, la necesidad implica el grado de deseabilidad de implementación del requisito, es decir, en caso de que hubiera que dejar alguno fuera por cuestiones de organización, los menos necesarios serían los excluidos. Su no redundancia se debe a que un requisito poco deseable puede tener una prioridad alta al tener requisitos cuya implementación dependan de este. Puede tomar los valores Esencial, Deseable u Opcional.
- **Descripción:** este campo se encarga de describir brevemente el motivo del requisito en forma de pequeño párrafo sin restricciones concretas que lo acoten.
- **Verificación:** cada uno de los requisitos debe ser verificado siguiendo los pasos expuestos en este apartado para comprobar su correcta implementación. El contenido de este apartado será un breve párrafo sin longitud acotada cuyo contenido indique los pasos para su correcta verificación.

Dados los pasos para la correcta especificación de requisitos, se muestran a continuación los mismos en el formato arriba indicado.

### *Requisitos funcionales*

#### **ID RF-001**

<b>Prioridad</b>	Alta
<b>Necesidad</b>	Esencial
<b>Descripción</b>	El programa debe permitir realizar un informe nuevo en todo momento
<b>Verificación</b>	Abrir el programa y realizar un informe Realizar un informe nuevo en caso de que ya se haya generado uno

Tabla 20: Requisito funcional RF-001

#### **ID RF-002**

<b>Prioridad</b>	Alta
<b>Necesidad</b>	Esencial
<b>Descripción</b>	El programa debe permitir realizar informes de la Demanda P48
<b>Verificación</b>	Realizar un nuevo informe del tipo Demanda P48

Tabla 21: Requisito funcional RF-002

#### **ID RF-003**

<b>Prioridad</b>	Alta
<b>Necesidad</b>	Esencial
<b>Descripción</b>	El programa debe permitir realizar informes de la Demanda PBF
<b>Verificación</b>	Realizar un nuevo informe del tipo Demanda PBF

Tabla 22: Requisito funcional RF-003

#### **ID RF-004**

<b>Prioridad</b>	Alta
<b>Necesidad</b>	Esencial
<b>Descripción</b>	El programa debe permitir realizar informes de la Demanda PVP
<b>Verificación</b>	Realizar un nuevo informe del tipo Demanda PVP

Tabla 23: Requisito funcional RF-004

**ID RF-005**

<b>Prioridad</b>	Alta
<b>Necesidad</b>	Esencial
<b>Descripción</b>	El programa debe permitir realizar informes del Programa P48
<b>Verificación</b>	Realizar un nuevo informe del tipo Programa P48

Tabla 24: Requisito funcional RF-005

**ID RF-006**

<b>Prioridad</b>	Alta
<b>Necesidad</b>	Esencial
<b>Descripción</b>	El programa debe permitir realizar informes del Programa PBF
<b>Verificación</b>	Realizar un nuevo informe del tipo Programa PBF

Tabla 25: Requisito funcional RF-006

**ID RF-007**

<b>Prioridad</b>	Alta
<b>Necesidad</b>	Esencial
<b>Descripción</b>	El programa debe permitir realizar informes del Programa PVP
<b>Verificación</b>	Realizar un nuevo informe del tipo Programa PVP

Tabla 26: Requisito funcional RF-007

**ID RF-008**

<b>Prioridad</b>	Alta
<b>Necesidad</b>	Esencial
<b>Descripción</b>	El programa debe permitir realizar informes de la Contratación PBF
<b>Verificación</b>	Realizar un nuevo informe del tipo Contratación PBF

Tabla 27: Requisito funcional RF-008

**ID RF-009**

<b>Prioridad</b>	Alta
<b>Necesidad</b>	Esencial
<b>Descripción</b>	El programa debe permitir realizar informes de la Previsión eólica
<b>Verificación</b>	Realizar un nuevo informe del tipo Previsión eólica

Tabla 28: Requisito funcional RF-009

**ID RF-010**

<b>Prioridad</b>	Alta
<b>Necesidad</b>	Esencial
<b>Descripción</b>	El programa debe permitir realizar informes de la Energía gestionada
<b>Verificación</b>	Realizar un nuevo informe del tipo Energía gestionada

Tabla 29: Requisito funcional RF-010

**ID RF-011**

<b>Prioridad</b>	Alta
<b>Necesidad</b>	Esencial
<b>Descripción</b>	El programa debe permitir realizar informes del Precio marginal
<b>Verificación</b>	Realizar un nuevo informe del tipo Precio marginal

Tabla 30: Requisito funcional RF-011

**ID RF-012**

<b>Prioridad</b>	Alta
<b>Necesidad</b>	Esencial
<b>Descripción</b>	El programa debe permitir realizar informes de distinto tipo con agregación diaria
<b>Verificación</b>	Realizar un nuevo informe de cualquier tipo con agregación diaria

Tabla 31: Requisito funcional RF-012

**ID RF-013**

<b>Prioridad</b>	Alta
<b>Necesidad</b>	Esencial
<b>Descripción</b>	El programa debe permitir realizar informes de distinto tipo con agregación mensual
<b>Verificación</b>	Realizar un nuevo informe de cualquier tipo con agregación mensual

Tabla 32: Requisito funcional RF-013

**ID RF-014**

<b>Prioridad</b>	Alta
<b>Necesidad</b>	Esencial
<b>Descripción</b>	El programa debe permitir realizar informes de distinto tipo con agregación anual
<b>Verificación</b>	Realizar un nuevo informe de cualquier tipo con agregación anual

Tabla 33: Requisito funcional RF-014

**ID RF-015**

<b>Prioridad</b>	Alta
<b>Necesidad</b>	Esencial
<b>Descripción</b>	El programa debe permitir realizar informes de distinto tipo con agregación interanual
<b>Verificación</b>	Realizar un nuevo informe de cualquier tipo con agregación interanual

Tabla 34: Requisito funcional RF-015

**ID RF-016**

<b>Prioridad</b>	Alta
<b>Necesidad</b>	Esencial
<b>Descripción</b>	El programa debe permitir realizar peticiones de distintos tipos de informes y agregaciones con rangos de fechas



## UNIVERSIDAD CARLOS III DE MADRID

<b>Verificación</b>	Realizar un nuevo informe de cualquier y agregación indicando fecha de inicio y de fin
---------------------	--

Tabla 35: Requisito funcional RF-016

**ID RF-017**

<b>Prioridad</b>	Alta
<b>Necesidad</b>	Esencial
<b>Descripción</b>	El programa debe permitir cambiar entre los informes generados por rangos de fechas
<b>Verificación</b>	Realizar un nuevo informe de cualquier y agregación indicando fecha de inicio y de fin y cambiar entre los distintos informes generados

Tabla 36: Requisito funcional RF-017

**ID RF-018**

<b>Prioridad</b>	Media
<b>Necesidad</b>	Deseable
<b>Descripción</b>	El programa no debe permitir realizar un informe sin antes especificar el tipo de informe deseado
<b>Verificación</b>	Intentar realizar una petición sin indicar el tipo de informe

Tabla 37: Requisito funcional RF-018

**ID RF-019**

<b>Prioridad</b>	Media
<b>Necesidad</b>	Deseable
<b>Descripción</b>	El programa no debe permitir realizar un informe sin antes especificar el tipo de agregación deseada
<b>Verificación</b>	Intentar realizar una petición sin indicar el tipo de agregación

Tabla 38: Requisito funcional RF-019

**ID RF-020**

<b>Prioridad</b>	Media
<b>Necesidad</b>	Deseable
<b>Descripción</b>	El programa no debe permitir realizar un informe sin antes especificar las fechas deseadas
<b>Verificación</b>	Intentar realizar una petición sin indicar el tipo de informe

Tabla 39: Requisito funcional RF-020

**ID RF-021**

<b>Prioridad</b>	Media
<b>Necesidad</b>	Esencial
<b>Descripción</b>	El programa no debe realizar un informe por intervalos si las fechas especificadas no lo precisan
<b>Verificación</b>	Realizar un informe normal y comprobar que no se muestre un intervalo

Tabla 40: Requisito funcional RF-021

## UNIVERSIDAD CARLOS III DE MADRID

<b>ID</b>	<b>RF-022</b>
<b>Prioridad</b>	Media
<b>Necesidad</b>	Esencial
<b>Descripción</b>	El programa no debe permitir cambiar entre informes si no se ha generado un informe por intervalos
<b>Verificación</b>	Realizar un informe normal y comprobar que no se permita cambiar entre informes

Tabla 41: Requisito funcional RF-022

<b>ID</b>	<b>RF-023</b>
<b>Prioridad</b>	Alta
<b>Necesidad</b>	Esencial
<b>Descripción</b>	El programa debe permitir guardar el informe completo realizado en formato adaptado al programa MatLab
<b>Verificación</b>	Realizar un informe de cualquier tipo y exportarlo en formato TXT

Tabla 42: Requisito funcional RF-023

<b>ID</b>	<b>RF-024</b>
<b>Prioridad</b>	Media
<b>Necesidad</b>	Deseable
<b>Descripción</b>	El programa debe permitir guardar el informe completo realizado en formato XML
<b>Verificación</b>	Realizar un informe de cualquier tipo y exportarlo en formato XML

Tabla 43: Requisito funcional RF-024

<b>ID</b>	<b>RF-025</b>
<b>Prioridad</b>	Media
<b>Necesidad</b>	Deseable
<b>Descripción</b>	El programa debe permitir guardar el informe completo realizado en formato CSV
<b>Verificación</b>	Realizar un informe de cualquier tipo y exportarlo en formato CSV

Tabla 44: Requisito funcional RF-025

<b>ID</b>	<b>RF-026</b>
<b>Prioridad</b>	Alta
<b>Necesidad</b>	Esencial
<b>Descripción</b>	El programa debe mostrar el informe solicitado en una tabla con formato adaptado al tipo y a la agregación
<b>Verificación</b>	Realizar un informe de cualquier tipo y comprobar que se muestra correctamente en la tabla

Tabla 45: Requisito funcional RF-026

<b>ID</b>	<b>RF-027</b>
<b>Prioridad</b>	Baja

## UNIVERSIDAD CARLOS III DE MADRID

<b>Necesidad</b>	Deseable
<b>Descripción</b>	El programa debe mostrar una gráfica generada a partir de los datos del informe solicitado adaptada al tipo y a la agregación
<b>Verificación</b>	Realizar un informe de cualquier tipo y comprobar que se muestra correctamente en la gráfica

Tabla 46: Requisito funcional RF-027

<b>ID</b>	<b>RF-028</b>
<b>Prioridad</b>	Baja
<b>Necesidad</b>	Opcional
<b>Descripción</b>	El programa debe poder mostrar información de la versión y el desarrollo
<b>Verificación</b>	Solicitar la información en el menú

Tabla 47: Requisito funcional RF-028

<b>ID</b>	<b>RF-029</b>
<b>Prioridad</b>	Media
<b>Necesidad</b>	Esencial
<b>Descripción</b>	El programa debe dar la posibilidad de salir del mismo en todo momento
<b>Verificación</b>	Salir del programa desde el menú o en el atajo de la ventana

Tabla 48: Requisito funcional RF-029

*Requisitos no funcionales*

<b>ID</b>	<b>RNF-001</b>
<b>Prioridad</b>	Alta
<b>Necesidad</b>	Esencial
<b>Descripción</b>	Para la ejecución del programa debe estar instalada la máquina virtual de Java en su versión 8 update 20 o superior
<b>Verificación</b>	Comprobar que está instalada antes de la ejecución del programa Instalar la máquina virtual de Java si fuera necesario

Tabla 49: Requisito no funcional RNF-001

<b>ID</b>	<b>RNF-002</b>
<b>Prioridad</b>	Media
<b>Necesidad</b>	Deseable
<b>Descripción</b>	El programa debe poder ser ejecutado en los principales sistemas operativos
<b>Verificación</b>	Ejecutar el programa desde distintos sistemas operativos compatibles con Java que cuenten con la máquina virtual ya instalada

Tabla 50: Requisito no funcional RNF-002

<b>ID</b>	<b>RNF-003</b>
<b>Prioridad</b>	Alta
<b>Necesidad</b>	Esencial
<b>Descripción</b>	El programa se debe ejecutar con una resolución mínima de 800x600 píxeles

<b>Verificación</b>	Ejecutar el programa con esa resolución o superior
---------------------	--

Tabla 51: Requisito no funcional RNF-003

**ID RNF-004**

<b>Prioridad</b>	Media
<b>Necesidad</b>	Deseable
<b>Descripción</b>	El programa debe mostrar una vista de bienvenida
<b>Verificación</b>	Abrir el programa y comprobar su existencia

Tabla 52: Requisito no funcional RNF-004

**ID RNF-005**

<b>Prioridad</b>	Alta
<b>Necesidad</b>	Esencial
<b>Descripción</b>	El programa debe contar con una barra superior de menú
<b>Verificación</b>	Ejecutar el programa y comprobar la existencia de la barra

Tabla 53: Requisito no funcional RNF-005

**ID RNF-006**

<b>Prioridad</b>	Alta
<b>Necesidad</b>	Esencial
<b>Descripción</b>	En la barra de menú debe existir la sección “Archivo”
<b>Verificación</b>	Abrir el programa y comprobar su existencia

Tabla 54: Requisito no funcional RNF-006

**ID RNF-007**

<b>Prioridad</b>	Media
<b>Necesidad</b>	Opcional
<b>Descripción</b>	En la barra de menú debe existir la sección “Ayuda”
<b>Verificación</b>	Abrir el programa y comprobar su existencia

Tabla 55: Requisito no funcional RNF-007

**ID RNF-008**

<b>Prioridad</b>	Alta
<b>Necesidad</b>	Esencial
<b>Descripción</b>	En el apartado “Archivo” de la barra de menú debe darse la posibilidad de realizar un nuevo informe
<b>Verificación</b>	Desplegar el menú “Archivo” y comprobar su existencia

Tabla 56: Requisito no funcional RNF-008

**ID RNF-009**

<b>Prioridad</b>	Media
<b>Necesidad</b>	Deseable

## UNIVERSIDAD CARLOS III DE MADRID

<b>Descripción</b>	En el apartado “Archivo” de la barra de menú debe darse la posibilidad de salir del programa
<b>Verificación</b>	Desplegar el menú “Archivo” y comprobar su existencia

Tabla 57: Requisito no funcional RNF-009

**ID RNF-010**

<b>Prioridad</b>	Baja
<b>Necesidad</b>	Opcional
<b>Descripción</b>	En el apartado “Ayuda” de la barra de menú debe existir la opción “Acerca de” donde se da información de la versión y el desarrollo del programa
<b>Verificación</b>	Desplegar el menú “Ayuda” y comprobar su existencia

Tabla 58: Requisito no funcional RNF-010

**ID RNF-011**

<b>Prioridad</b>	Alta
<b>Necesidad</b>	Esencial
<b>Descripción</b>	El programa debe contar con una vista en la que se permita realizar informes y visualizarlos
<b>Verificación</b>	Abrir el programa y pulsar la opción “Nuevo informe”

Tabla 59: Requisito no funcional RNF-011

**ID RNF-012**

<b>Prioridad</b>	Alta
<b>Necesidad</b>	Esencial
<b>Descripción</b>	En la vista de informes deben existir seleccionables para el tipo de informe, la agregación y las fechas
<b>Verificación</b>	Abrir el programa y pulsar la opción “Nuevo informe”, comprobar entonces que esas opciones existen

Tabla 60: Requisito no funcional RNF-012

**ID RNF-013**

<b>Prioridad</b>	Alta
<b>Necesidad</b>	Esencial
<b>Descripción</b>	En la vista de informes deben existir las opciones de realizar y exportar informes
<b>Verificación</b>	Abrir el programa y pulsar la opción “Nuevo informe”, comprobar entonces que esas opciones existen

Tabla 61: Requisito no funcional RNF-013

**ID RNF-014**

<b>Prioridad</b>	Alta
<b>Necesidad</b>	Esencial
<b>Descripción</b>	En la vista de informes deben existir las opciones de cambiar entre informe en caso de que se haya realizado una consulta por rango de fechas

## UNIVERSIDAD CARLOS III DE MADRID

<b>Verificación</b>	Abrir el programa y pulsar la opción “Nuevo informe”, realizar una consulta por rango de fechas y comprobar que esta opción existe
---------------------	--

Tabla 62: Requisito no funcional RNF-014

**ID RNF-015**

<b>Prioridad</b>	Alta
<b>Necesidad</b>	Esencial
<b>Descripción</b>	En la vista de informes debe existir una tabla en la que representar el informe
<b>Verificación</b>	Abrir el programa y pulsar la opción “Nuevo informe” y comprobar que existe

Tabla 63: Requisito no funcional RNF-015

**ID RNF-016**

<b>Prioridad</b>	Alta
<b>Necesidad</b>	Esencial
<b>Descripción</b>	En la vista de informes debe existir una gráfica en la que representar el informe
<b>Verificación</b>	Abrir el programa y pulsar la opción “Nuevo informe” y comprobar que existe

Tabla 64: Requisito no funcional RNF-016

**ID RNF-017**

<b>Prioridad</b>	Baja
<b>Necesidad</b>	Opcional
<b>Descripción</b>	En la vista de informes se debe poder redimensionar la vista de tabla y la gráfica en función del gusto del usuario
<b>Verificación</b>	Abrir el programa y pulsar la opción “Nuevo informe” y comprobar que se puede redimensionar la vista

Tabla 65: Requisito no funcional RNF-017

**ID RNF-018**

<b>Prioridad</b>	Baja
<b>Necesidad</b>	Opcional
<b>Descripción</b>	La ventana del programa debe poder redimensionarse
<b>Verificación</b>	Abrir el programa y comprobar que es redimensionable

Tabla 66: Requisito no funcional RNF-018

**ID RNF-019**

<b>Prioridad</b>	Media
<b>Necesidad</b>	Deseable
<b>Descripción</b>	El diseño de la ventana y los elementos de la aplicación debe ser coherente con el sistema operativo en el que se está ejecutando
<b>Verificación</b>	Abrir el programa en distintos sistemas operativos y comprobar la coherencia del diseño

Tabla 67: Requisito no funcional RNF-019

<b>ID RNF-020</b>	
<b>Prioridad</b>	Alta
<b>Necesidad</b>	Esencial
<b>Descripción</b>	El programa se dividirá a nivel de arquitectura siguiendo un patrón clásico de modelo vista controlador
<b>Verificación</b>	Abrir el proyecto y comprobar la estructura de paquetes de clases

Tabla 68: Requisito no funcional RNF-020

<b>ID RNF-021</b>	
<b>Prioridad</b>	Alta
<b>Necesidad</b>	Esencial
<b>Descripción</b>	El programa realizará peticiones a un servidor ajeno al proyecto mediante peticiones HTTP
<b>Verificación</b>	Abrir el programa y realizar una petición

Tabla 69: Requisito no funcional RNF-021

<b>ID RNF-022</b>	
<b>Prioridad</b>	Alta
<b>Necesidad</b>	Esencial
<b>Descripción</b>	El programa no debe permitir al usuario realizar acciones incoherentes tales como pedir rangos de fecha cuya fecha final sea inferior a la inicial
<b>Verificación</b>	Abrir el programa e intentar realizar una petición incoherente

Tabla 70: Requisito no funcional RNF-022

### Casos de uso

Los requisitos se crean para definir el desarrollo de las herramientas que permiten la correcta interacción entre un usuario cualquiera y el sistema que se está desarrollando en el proyecto. Esta interacción genera casos de uso concretos que deben ser descritos con propiedad y siguiendo una plantilla que sea capaz de clasificarlos a todos. Al igual que ocurre con la plantilla de requisitos, se muestra a continuación la tabla 71 que será usada como la plantilla correspondiente a los casos de uso del sistema:

<b>ID</b>	
<b>Nombre</b>	
<b>Actores</b>	
<b>Importancia</b>	
<b>Descripción</b>	
<b>Precondiciones</b>	
<b>Flujo de eventos</b>	

Tabla 71: Plantilla para los casos de uso

Para su mejor comprensión, se explican aquí abajo todos los campos de los que se compone la plantilla de los casos de uso:

- **ID:** al igual que en el caso de los requisitos, los casos de uso también cuentan con un identificador único que no puede ser repetido. En este caso, se forma empezando con las siglas de caso de uso (CU), seguido de un guion y un código numérico de tres cifras único para cada uno. El código del primer caso de uso sería CU-001 y se incrementaría por unidades para cada uno de los casos de uso siguientes.
- **Nombre:** frase breve descriptiva sobre el caso de uso.
- **Actores:** tipo de usuario del sistema objetivo del caso de uso concreto.
- **Importancia:** este campo indica el grado de importancia del caso de uso dentro del proyecto. Existen tres grados distintos de importancia; Alta, Media y Baja.
- **Descripción:** en este campo se describe brevemente el caso de uso en forma de pequeño párrafo sin restricciones concretas que lo acoten más allá de la longitud razonable de un párrafo.
- **Precondiciones:** en las precondiciones se describen las condiciones necesarias para que se pueda dar el caso de uso en caso de que existan.
- **Flujo de eventos:** aquí se describe paso por paso las acciones que componen el caso de uso concreto, es decir, los pasos que debe realizar el actor para concluir satisfactoriamente el caso de uso que se está tratando. En caso de que existiera más de una vía para llegar al final del caso de uso, debe explicarse también en este campo de la tabla.

A continuación se exponen los casos de uso del sistema usando la plantilla descrita con anterioridad. Puesto que crear un caso de uso para cada tipo de informe y agregación haría extenderse la lista creando redundancias innecesarias, se ha tratado de simplificar los casos de uso al máximo creando casos de uso genéricos.

ID CU-001	
<b>Nombre</b>	Crear informe cualquiera
<b>Actores</b>	Usuario
<b>Importancia</b>	Alta
<b>Descripción</b>	El usuario necesita crear un informe de cualquier tipo y agregación
<b>Precondiciones</b>	El usuario debe contar con la máquina virtual de Java instalada y el programa en su equipo además de contar con conexión a internet
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El usuario abre el programa</li> <li>2. El usuario pulsa en el menú “Nuevo informe” dentro del apartado “Archivo”</li> <li>3. El usuario selecciona el tipo de informe deseado</li> <li>4. El usuario selecciona el tipo de agregación deseada</li> <li>5. El usuario selecciona la fecha deseada para el informe</li> <li>6. El usuario pulsa el botón “Consultar”</li> </ol>

Tabla 72: Caso de uso CU-001

ID CU-002	
<b>Nombre</b>	Crear informe cualquiera por rangos de fechas
<b>Actores</b>	Usuario
<b>Importancia</b>	Alta
<b>Descripción</b>	El usuario necesita crear un informa de cualquier tipo y agregación con un rango concreto de fechas



<b>Precondiciones</b>	El usuario debe contar con la máquina virtual de Java instalada y el programa en su equipo además de contar con conexión a internet
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El usuario abre el programa</li> <li>2. El usuario pulsa en el menú “Nuevo informe” dentro del apartado “Archivo”</li> <li>3. El usuario selecciona el tipo de informe deseado</li> <li>4. El usuario selecciona el tipo de agregación deseada</li> <li>5. El usuario selecciona las fechas deseadas para el informe acotando fecha inicial y final</li> <li>6. El usuario pulsa el botón “Consultar”</li> </ol>

Tabla 73: Caso de uso CU-002

**ID CU-003**

<b>Nombre</b>	Visualizar datos de un informe cualquiera en forma de tabla
<b>Actores</b>	Usuario
<b>Importancia</b>	Media
<b>Descripción</b>	El usuario necesita visualizar los datos de un informe de cualquier tipo, agregación y rango de fechas en forma de tabla
<b>Precondiciones</b>	El usuario debe contar con la máquina virtual de Java instalada y el programa en su equipo además de contar con conexión a internet o el usuario ha creado con anterioridad un informe
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El usuario abre el programa</li> <li>2. El usuario pulsa en el menú “Nuevo informe” dentro del apartado “Archivo”</li> <li>3. El usuario selecciona el tipo de informe deseado</li> <li>4. El usuario selecciona el tipo de agregación deseada</li> <li>5. El usuario selecciona las fechas deseadas para el informe acotando fecha inicial y final</li> <li>6. El usuario pulsa el botón “Consultar”</li> <li>7. La tabla de la vista refleja el informe deseado por el usuario</li> </ol> <p style="text-align: center;">o</p> <ol style="list-style-type: none"> <li>1. El informe se encuentra creado</li> <li>2. La tabla de la vista refleja el informe deseado por el usuario</li> </ol>

Tabla 74: Caso de uso CU-003

**ID CU-004**

<b>Nombre</b>	Visualizar datos de un informe cualquiera en forma de gráfica
<b>Actores</b>	Usuario
<b>Importancia</b>	Media
<b>Descripción</b>	El usuario necesita visualizar los datos de un informe de cualquier tipo, agregación y rango de fechas en forma de gráfica
<b>Precondiciones</b>	El usuario debe contar con la máquina virtual de Java instalada y el programa en su equipo además de contar con conexión a internet o el usuario ha creado con anterioridad un informe
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El usuario abre el programa</li> </ol>

	<ol style="list-style-type: none"> <li>2. El usuario pulsa en el menú “Nuevo informe” dentro del apartado “Archivo”</li> <li>3. El usuario selecciona el tipo de informe deseado</li> <li>4. El usuario selecciona el tipo de agregación deseada</li> <li>5. El usuario selecciona las fechas deseadas para el informe acotando fecha inicial y final</li> <li>6. El usuario pulsa el botón “Consultar”</li> <li>7. La gráfica de la vista refleja el informe deseado por el usuario</li> </ol> <p style="text-align: center;">o</p> <ol style="list-style-type: none"> <li>1. El informe se encuentra creado</li> <li>2. La gráfica de la vista refleja el informe deseado por el usuario</li> </ol>
--	---

Tabla 75: Caso de uso CU-004

<b>ID CU-005</b>	
<b>Nombre</b>	Guardar un informe cualquiera
<b>Actores</b>	Usuario
<b>Importancia</b>	Alta
<b>Descripción</b>	El usuario necesita guardar un informe de cualquier tipo, agregación y rango de fechas en cualquier tipo de formato
<b>Precondiciones</b>	El usuario debe contar con la máquina virtual de Java instalada y el programa en su equipo además de contar con conexión a internet o el usuario ha creado con anterioridad un informe
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El usuario abre el programa</li> <li>2. El usuario pulsa en el menú “Nuevo informe” dentro del apartado “Archivo”</li> <li>3. El usuario selecciona el tipo de informe deseado</li> <li>4. El usuario selecciona el tipo de agregación deseada</li> <li>5. El usuario selecciona las fechas deseadas para el informe acotando fecha inicial y final</li> <li>6. El usuario pulsa el botón “Consultar”</li> <li>7. El usuario pulsa el botón “Exportar informe”</li> <li>8. El usuario selecciona el tipo de archivo</li> <li>9. El usuario selecciona la ruta de guardado</li> <li>10. El usuario selecciona el nombre del archivo</li> <li>11. El usuario pulsa el botón “Guardar”</li> </ol> <p style="text-align: center;">o</p> <ol style="list-style-type: none"> <li>12. El informe se encuentra creado</li> <li>13. El usuario pulsa el botón “Exportar informe”</li> <li>14. El usuario selecciona el tipo de archivo</li> <li>15. El usuario selecciona la ruta de guardado</li> <li>16. El usuario selecciona el nombre del archivo</li> <li>17. El usuario pulsa el botón “Guardar”</li> </ol>

Tabla 76: Caso de uso CU-005

ID CU-006	
<b>Nombre</b>	Ver la información del programa
<b>Actores</b>	Usuario
<b>Importancia</b>	Baja
<b>Descripción</b>	El usuario desea ver la información del programa
<b>Precondiciones</b>	El usuario debe contar con la máquina virtual de Java instalada y el programa en su equipo
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El usuario abre el programa</li> <li>2. El usuario pulsa en el menú “Acerca de” dentro del apartado “Ayuda”</li> <li>3. El programa muestra la información deseada</li> </ol>

Tabla 77: Caso de uso CU-006

ID CU-007	
<b>Nombre</b>	Salir del programa
<b>Actores</b>	Usuario
<b>Importancia</b>	Media
<b>Descripción</b>	El usuario ha realizado todas las operaciones que necesitaba y desea salir del programa
<b>Precondiciones</b>	El usuario debe tener el programa abierto
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa en el menú “Salir” dentro del apartado “Archivo”</li> <li>2. El programa se cierra</li> </ol> <p style="text-align: center;">o</p> <ol style="list-style-type: none"> <li>1. El usuario pulsa en el botón de cerrar de la interfaz</li> <li>2. El programa se cierra</li> </ol>

Tabla 78: Caso de uso CU-007

### Matriz de trazabilidad

La matriz de trazabilidad es una tabla que representa la relación entre los requisitos definidos y los casos de uso que derivan de estos, es decir, de que requisitos derivan los casos de uso. Para la realización de esta matriz, sólo se tienen en cuenta los requisitos funcionales ya que los no funcionales por definición no pueden representar casos de uso al centrarse en aspectos más etéreos. La tabla 79 muestra el cruce entre los casos de uso (en el eje horizontal) y los requisitos funcionales (en el eje vertical):

ID	CU-001	CU-002	CU-003	CU-004	CU-005	CU-006	CU-007
RF-001	●	●					
RF-002	●	●					
RF-003	●	●					
RF-004	●	●					
RF-005	●	●					
RF-006	●	●					
RF-007	●	●					
RF-008	●	●					

RF-009	●	●					
RF-010	●	●					
RF-011	●	●					
RF-012	●	●					
RF-013	●	●					
RF-014	●	●					
RF-015	●	●					
RF-016		●					
RF-017		●					
RF-018	●	●					
RF-019	●	●					
RF-020	●	●					
RF-021	●	●					
RF-022		●					
RF-023					●		
RF-024					●		
RF-025					●		
RF-026			●				
RF-027				●			
RF-028						●	
RF-029							●

Tabla 79: Matriz de trazabilidad del sistema

## Diseño

El diseño es la fase del desarrollo que sigue al análisis. En esta fase ya se han establecido las necesidades y requisitos del proceso de desarrollo, para a continuación definir las guías necesarias para la fase de implementación del programa mediante la construcción de modelos y representaciones gráficas además de las explicaciones necesarias para llevar a cabo correctamente la fase de programación.

La fase de diseño se ha atravesado distintas veces mientras se seguía el modelo de proceso usado, una metodología tradicional con retroalimentación, lo que implica que se puede volver atrás en las fases del desarrollo para corregir posibles errores o añadir mejoras. De esta forma la información mostrada en este apartado ha sido depurada en fases posteriores, viendo la coherencia y la necesidad de según qué cosas (clases innecesarias, métodos auxiliares deseables, etc.) sobre todo durante la fase de implementación.

### Diseño del sistema

La arquitectura aplicada para el diseño de la solución es la definida como Modelo Vista Controlador [MVC]. El objetivo de esta arquitectura es abstraer la parte de modelado de la aplicación de la parte de interfaz de manera que ambos desarrollos puedan seguir de forma paralela. La tercera capa, el controlador, es la parte lógica encargada de unir las dos primeras de forma que todo funcione correctamente independientemente de la forma en la que hayan continuado ambos desarrollos. En la figura 19 se puede apreciar el esquema clásico de este tipo de arquitectura.

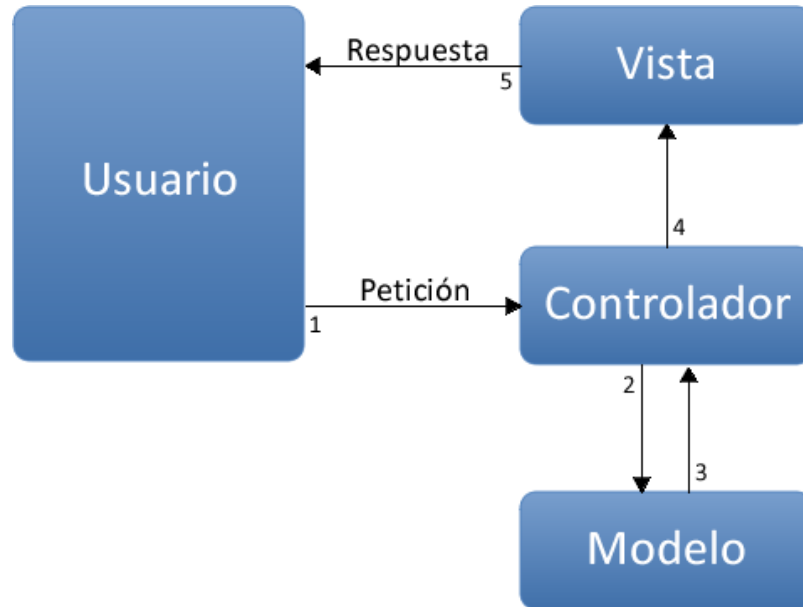


Figura 19: Esquema de la arquitectura Modelo Vista Controlador

Más en detalle, se puede decir que cada una de estas tres capas se encarga de cubrir los siguientes elementos:

- **Modelo:** esta primera capa es en la que se define el modelo, la forma o los tipos de datos de la aplicación. Al no precisar parte de servidor ni persistencia de datos propiamente dicha (los informes pueden guardarse pero la aplicación no precisa datos que se guarden entre sesiones para su correcto funcionamiento), se pueden obviar modelos de bases de datos y se deben realizar en el lenguaje en el que se programe finalmente la aplicación, por lo que esta parte será programada en Java. En ella se definen todos los objetos de Java que modelan la aplicación como pueden ser la petición y la respuesta de los informes de tal forma que queden definidos de cara a su uso en la vista.
- **Vista:** la parte de la vista está realizada en una librería relativamente nueva, se trata de JavaFX. Como ya se ha explicado con anterioridad, esta librería se basa en una sintaxis de tipo XML (de ahí que la extensión de los archivos sea .fxml) con etiquetas propias correspondientes a tipos de objetos modelados por esta interfaz. Su alta integración con Java y las herramientas existentes para su desarrollo la hacen la mejor candidata para su uso, además de los resultados atractivos que son posibles alcanzar con poca experiencia.
- **Controlador:** en esta última capa se encuentra toda la lógica del programa y actúa a su vez como puente entre las capas de modelo y de vista. Todas las gestiones y acciones que se pueden hacer mediante la vista, se realizan en esta capa afectando elementos de la capa de modelo. Su desarrollo se realiza en el lenguaje de la aplicación, es decir, en Java.

Como se puede observar, para un flujo de eventos normal en el uso del programa, el usuario mediante la vista, ejecutará acciones actuando sobre el controlador, que a su vez usará el modelo, para más tarde actualizar la vista con los datos pertinentes.

La comunicación entre las tres capas se realiza íntegramente en lenguaje Java al no precisar de componentes externos como bases de datos.

### *Diseño detallado*

Habiendo comentado ya en que se basa cada una de las capas del diseño de la aplicación, queda explicar en detalle cada uno de los componentes del programa contenidos en estas tres capas de abstracción que se han decidido usar.

### *Capa de modelo*

El modelo equivaldría a la capa de persistencia del proyecto si hubiera sido definido en una base de datos. En este caso, al no necesitar persistencia debido a que su funcionamiento se basa en el acceso a los datos únicamente mediante peticiones a la web de Red Eléctrica, la capa de modelo está hecha únicamente en Java. Si bien se podría decir que la parte de guardado de archivos equivale a la parte de persistencia del programa, este guardado es para su uso y procesado posterior, no para el correcto funcionamiento de la herramienta.

La capa de modelo cuenta con tres clases de Java que representan las peticiones, las respuestas y los tipos de archivos soportados por el programa. Las tres clases quedan definidas en las tablas siguientes:

- **Clase Archivo**

La clase Archivo modela los tipos de archivo disponibles a la hora de guardar un informe ya solicitado.

Clase Archivo.java			
Atributos	Nombre	Tipo	Descripción
	ALL	ExtensionFilter	Constante usada a la hora de guardar un archivo para admitir todos los tipos
	TXT	ExtensionFilter	Constante usada a la hora de guardar un archivo para filtrar únicamente por archivos TXT usados por MatLab
	CSV	ExtensionFilter	Constante usada a la hora de guardar un archivo para filtrar únicamente por archivos CSV usados

Tabla 80: Clase Archivo

- **Clase Peticion**

La clase Petición modela los datos relativos a esta para que se pueda realizar correctamente.

Clase Peticion.java			
Constantes	Nombre	Tipo	Descripción
	DEMANDA_P48	int	Constante usada para referirse a las peticiones

		de informes del tipo Demanda P48
PROGRAMA_P48	int	Constante usada para referirse a las peticiones de informes del tipo Programa P48
DEMANDA_PBF	int	Constante usada para referirse a las peticiones de informes del tipo Demanda PBF
PROGRAMA_PBF	int	Constante usada para referirse a las peticiones de informes del tipo Programa PBF
DEMANDA_PVP	int	Constante usada para referirse a las peticiones de informes del tipo Demanda PVP
PROGRAMA_PVP	int	Constante usada para referirse a las peticiones de informes del tipo Programa PVP
CONTRATACION_PBF	int	Constante usada para referirse a las peticiones de informes del tipo Contratación PBF
PREV_EOLICA	int	Constante usada para referirse a las peticiones de informes del tipo Previsión eólica
ENERGIA	int	Constante usada para referirse a las peticiones de informes del tipo Energía gestionada
PRECIO	int	Constante usada para referirse a las peticiones de informes del tipo Precios marginales
AGR_NO_SELECCIONADA	int	Constante usada para referirse a una agregación no seleccionada
AGR_DIARIA	Int	Constante usada para referirse a un informe con una agregación diaria seleccionada

<b>Atributos</b>	AGR_MENSUAL	Int	Constante usada para referirse a un informe con una agregación mensual seleccionada
	AGR_ANUAL	Int	Constante usada para referirse a un informe con una agregación anual seleccionada
	AGR_INTERANUAL	Int	Constante usada para referirse a un informe con una agregación interanual seleccionada
	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	nombreBase	String	Atributo que describe el tipo de informe que se está solicitando en la petición
	nombreFichero	String	El nombre del fichero está comprendido por una abreviatura del tipo informe solicitado, una abreviatura del tipo de agregación y la fecha de la petición si no es de tipo interanual
	agregacion	String	Abreviatura de la agregación solicitada
	fechaSolicitada	String	Fecha solicitada
	agregacionSolicitada	String	Agregación solicitada
	limiteInferior	String	Por defecto vacío, posiblemente se introdujera para solicitar más de un informe
	limiteSuperior	String	Fecha de la petición
	limiteSuperiorBase	String	Fecha de la petición
	baseGrafica	String	Por defecto vacío, posiblemente tuviera que ver con lo mismo que el limiteInferior
	nombreGrafico	String	Nombre de la gráfica, si no se especifica se aplica por defecto
	filtro	String	Abreviatura del filtro solicitado, si no se especifica no se aplica ninguno



<b>Métodos</b>	filtroSolicitado	String	Filtro solicitado
	radioSolicitado	String	Por defecto vacío
	idiomaSeleccionado	String	Código de idioma solicitado, si no se especifica el idioma es español
	graficaGoogle	String	Por defecto vacío
	tiPO	String	Tipo de
	fECHA	String	Fecha de la petición
	<b>Nombre</b>	<b>Retorno / Argumentos</b>	<b>Descripción</b>
	Peticion	Constructor	Constructor por defecto de la clase
	Peticion	Constructor String nombreBase String nombreFichero String agregacion String fechaSolicitada String agregacionSolicitada String limiteInferior String limiteSuperior String limiteSuperiorBase String baseGrafica String nombreGrafico String filtro String filtroSolicitado String radioSolicitado String idiomaSeleccionado String graficaGoogle String tiPO String fECHA	Constructor completo de la clase
	getNombreBase	String	Método que retorna el atributo nombreBase
	setNombreBase	void String nombreBase	Método que establece el atributo de la clase nombreBase con el que recibe
	getNombreFichero	String	Método que retorna el atributo nombreFichero
	setNombreFichero	void String nombreFichero	Método que establece el atributo de la clase nombreFichero con el que recibe
	getAgregacion	String	Método que retorna el atributo agregación

setAgregacion	void String agregacion	Método que establece el atributo de la clase agregación con el que recibe
getFechaSolicitada	String	Método que retorna el atributo fechaSolicitada
setFechaSolicitada	void String fechaSolicitada	Método que establece el atributo de la clase fechaSolicitada con el que recibe
getAgregacionSolicitada	String	Método que retorna el atributo agregacionSolicitada
setAgregacionSolicitada	void String agregacionSolicitada	Método que establece el atributo de la clase agregacionSolicitada con el que recibe
getLimiteInferior	String	Método que retorna el atributo limiteInferior
setLimiteInferior	void String limiteInferior	Método que establece el atributo de la clase limiteInferior con el que recibe
getLimiteSuperior	String	Método que retorna el atributo limiteSuperior
setLimiteSuperior	void String limiteSuperior	Método que establece el atributo de la clase limiteSuperior con el que recibe
getLimiteSuperiorBase	String	Método que retorna el atributo limiteSuperiorBase
setLimiteSuperiorBase	void String limiteSuperiorBase	Método que establece el atributo de la clase limiteSuperiorBase con el que recibe
getBaseGrafica	String	Método que retorna el atributo baseGrafica
setBaseGrafica	void String baseGrafica	Método que establece el atributo de la clase baseGrafica con el que recibe
getNombreGrafico	String	Método que retorna el atributo nombreGrafica
setNombreGrafico	void String nombreGrafico	Método que establece el atributo de la clase

		nombreGrafica con el que recibe
getFiltro	String	Método que retorna el atributo filtro
setFiltro	void String filtro	Método que establece el atributo de la clase filtro con el que recibe
getFiltroSolicitado	String	Método que retorna el atributo filtroSolicitado
setFiltroSolicitado	void String filtroSolicitado	Método que establece el atributo de la clase filtroSolicitado con el que recibe
getRadioSolicitado	String	Método que retorna el atributo radioSolicitado
setRadioSolicitado	void String radioSolicitado	Método que establece el atributo de la clase radioSolicitado con el que recibe
getIdiomaSeleccionado	String	Método que retorna el atributo idiomaSeleccionado
setIdiomaSeleccionado	void String idiomaSeleccionado	Método que establece el atributo de la clase idiomaSeleccionado con el que recibe
getGraficaGoogle	String	Método que retorna el atributo graficaGoogle
setGraficaGoogle	void String graficaGoogle	Método que establece el atributo de la clase graficaGoogle con el que recibe
getTIPO	String	Método que retorna el atributo tIPO
setTIPO	void String tIPO	Método que establece el atributo de la clase tIPO con el que recibe
getFECHA	String	Método que retorna el atributo fECHA
setFECHA	void String fECHA	Método que establece el atributo de la clase fECHA con el que recibe

Tabla 81: Clase Peticion

En esta clase se han reflejado todos los parámetros que tiene una petición completa a los servicios de Red eléctrica tal y como vienen descritos en los ejemplos encontrados. Aunque como

se puede observar algunos de los atributos son redundantes y a la hora de la realizar la petición sólo son necesarios algunos puntos clave, se ha decidido modelar por completo para evitar posibles errores futuros.

- **Clase Respuesta**

La clase respuesta es la encargada de modelar los informes una vez recibidos de tal forma que el programa los pueda manejar correctamente en todos los pasos, desde mostrarlos hasta exportarlos.

<b>Clase Respuesta.java</b>			
<b>Atributos</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	tipoPeticion	int	Atributo que describe el tipo de informe modelado en la respuesta
	tipoAgregacion	int	Atributo que indica el tipo de agregación del informe
	fecha	LocalDate	Fecha del informe
	nombreDatos	String[]	Array de nombres de los parámetros del informe
	tiempoDatos	String[]	Array compuesto por los parámetros temporales de la respuesta, es decir, si es diaria, las horas, si es mensual, los días, etc.
	datos	Double[][]	Matriz de datos de la respuesta
<b>Métodos</b>	<b>Nombre</b>	<b>Retorno / Argumentos</b>	<b>Descripción</b>
	Respuesta	Constructor	Constructor por defecto de la clase
	Respuesta	Constructor LocalDate fecha int tipoPeticion int tipoAgregacion	Constructor con información inicial de la clase
	getTipoPeticion	int	Método que retorna el atributo tipoPeticion
	setTipoPeticion	void int tipoPeticion	Método que establece el atributo de la clase tipoPeticion con el que recibe
	getTipoAgregacion	String	Método que retorna el atributo tipoAgregacion
	setTipoAgregacion	void int tipoAgregacion	Método que establece el atributo de la clase tipoAgregacion con el que recibe
	getFecha	LocalDate	Método que retorna el atributo fecha

setFecha	void LocalDate fecha	Método que establece el atributo de la clase fecha con el que recibe
getNombreDatos	String[]	Método que retorna el atributo nombreDatos
setNombreDatos	void String[] nombreDatos	Método que establece el atributo de la clase nombreDatos con el que recibe
getTiempoDatos	String[]	Método que retorna el atributo tiempoDatos
setTiempoDatos	void String[] tiempoDatos	Método que establece el atributo de la clase tiempoDatos con el que recibe
getDatos	Double[][]	Método que retorna el atributo datos
setDatos	void Double[][] datos	Método que establece el atributo de la clase datos con el que recibe
getDato	Double int x int y	Método que retorna un valor concreto de la matriz completa de datos

Tabla 82: Clase Respuesta

Esta clase está hecha de tal forma que sea capaz de aceptar cualquier tipo de respuesta generada a partir de cualquier petición. De esta forma se consigue evitar redundancias de código al crear un objeto respuesta genérico que modele todos los aspectos.

### Capa de vista

En la capa de vista se explican los componentes de las perspectivas que componen la interfaz del programa. Al estar construida con JavaFX que cuenta con una sintaxis XML, los componentes pueden contener otros componentes a su vez. Con el fin de explicar los elementos de la capa de vista con total claridad, a cada uno de los componentes se le asignará un ID único dentro de su perspectiva para poder referenciarlos con facilidad. El ID estará formado por tres caracteres numéricos que se asignarán en orden ascendente.

- **Vista Inicial**

La vista Inicial es la vista de bienvenida en la que se muestra un diagrama del funcionamiento de la aplicación y una breve explicación sobre cómo proceder con la realización de informes. Se inserta dentro de la vista Principal para formar la perspectiva de bienvenida.

#### Vista VistalInicial.fxml

ID	Elemento	Contiene/Contenido en	Descripción
001	AnchorPane	Contiene todos los elementos de la vista	Elemento contenedor donde se encuentran todos los elementos de esta vista

002	ImageView	Contenido en el AnchorPane 001	Elemento para insertar imágenes en una vista. Contiene un esquema descriptivo del funcionamiento de la herramienta
003	Text	Contenido en el AnchorPane 001	Elemento para reflejar texto. Contiene el texto “¡Bienvenido!”
004	Text	Contenido en el AnchorPane 001	Elemento para reflejar texto. Contiene el texto “Para comenzar selecciona “Nuevo informe” en el menú “Archivo””

Tabla 83: Vista Inicial

- **Vista Principal**

La vista Principal es la que enmarca el resto de vistas de la aplicación. Se compone del menú superior, una barra inferior y el marco donde se deben insertar el resto de vistas. Forma parte de todas las perspectivas.

**Vista VistaPrincipal.fxml**

ID	Elemento	Contiene/Contenido en	Descripción
001	BorderPane	Contiene todos los elementos de la vista	El BorderPane es un elemento contenedor que separa la vista por secciones (los cuatro bordes y un área central). Dentro de este elemento se encuentran todos los elementos de la interfaz. En su área central se insertan el resto de vistas mediante los controladores anteriormente explicados
002	AnchorPane	Contiene el menú de la aplicación Contenido en la parte superior del BorderPane 001	El elemento AnchorPane es otro tipo de elemento contenedor. Este en concreto contiene el menú superior de la aplicación
003	MenuBar	Contiene los elementos del menú de la aplicación Contenido en el AnchorPane 002	El MenuBar es una barra tradicional de menú. Contiene los elementos emergentes tradicionales asociados a este tipo de barras
004	Menu	Contiene los elementos del menú de Archivo Contenido en la MenuBar 003	Cada uno de los elementos Menu modelan una sección en la barra de menú. Este en concreto se refiere a la sección “Archivo”
005	Menu	Contiene los elementos del menú de Ayuda	Este elemento Menu modela la sección “Ayuda”

		Contenido en la MenuBar 003	
006	MenuItem	Contenido en el Menu 004	Este elemento modela una entrada en el menú de la aplicación. Este en concreto se refiere al elemento “Nuevo informe”
007	SeparatorMenuItem	Contenido en el Menu 004	Este elemento crea una barra separadora dentro de un menú emergente
008	MenuItem	Contenido en el Menu 004	Este elemento modela la opción “Salir” dentro del menú “Archivo”
009	MenuItem	Contenido en el Menu 005	Este elemento modela la opción “Acerca de” dentro del menú “Ayuda”
010	AnchorPane	Contiene los textos inferiores de la ventana Contenido en la parte inferior del BorderPane 001	Este elemento contenedor cuenta con dos textos para la barra inferior de la aplicación
011	Text	Contenido en el AnchorPane 010	Modela el texto de título del programa mostrado en la barra inferior “Red Eléctrica Web Scraper - <a href="http://www.esios.ree.es/web-publica/">http://www.esios.ree.es/web-publica/</a> ”
012	Text	Contenido en el AnchorPane 010	Modela el nombre de la universidad mostrado en la barra inferior “Universidad Carlos III de Madrid”

Tabla 84: Vista Principal

- **Vista Respuesta**

La vista Respuesta es la más importante pues es en la que se realizan los informes, desde la que se visualizan y desde la que se exportan. Contiene todos los componentes necesarios para poder realizar una petición y visualizar la respuesta correctamente. Está contenida dentro de la vista Principal formando la perspectiva de respuesta.

**Vista VistaRespuesta.fxml**

ID	Elemento	Contiene/Contenido en	Descripción
001	AnchorPane	Contiene todos los elementos de la vista	Es un elemento contenedor que alberga todos los elementos de esta vista
002	ToolBar	Contiene el menú de petición de la vista	Elemento típico asociado a una barra de herramientas que

		Contenido en el AnchorPane 001	contiene las opciones para la petición del informe
003	ToolBar	Contiene el menú de exportación y cambio de informe si se ha solicitado un rango de fechas Contenido en el AnchorPane 001	Elemento típico asociado a una barra de herramientas que contiene las opciones de exportación y cambio de informe
004	SplitPane	Contiene las tablas y gráficas en las que se muestra el informe Contenido en el AnchorPane 001	Elemento que aporta la posibilidad de redimensionar internamente las dos áreas que crea
005	ChoiceBox	Contenido en la ToolBar 002	Selector que muestra los tipos de informe
006	ChoiceBox	Contenido en la ToolBar 002	Selector que muestra los tipos de agregación
007	Text	Contenido en la ToolBar 002	Texto que modela la cadena "Fechas"
008	DatePicker	Contenido en la ToolBar 002	Selector de la fecha inicial
009	DatePicker	Contenido en la ToolBar 002	Selector de la fecha final
010	Button	Contenido en la ToolBar 002	Botón con el texto "Generar informe" que realiza la petición si los datos están correctamente rellenos
011	Button	Contenido en la ToolBar 002	Botón con el texto "Exportar informe" que abre el cuadro de diálogo del sistema para el guardado del informe realizado
012	Button	Contenido en la ToolBar 003	Botón con el texto "< Anterior" que retrocede un informe en caso de que se haya realizado una petición por rango de fechas y sea posible
013	TextField	Contenido en la ToolBar 003	Campo de texto que muestra la fecha del informe mostrado
014	Button	Contenido en la ToolBar 003	Botón con el texto "Siguiente >" que avanza un informe en caso de que se haya realizado una petición por rango de fechas y sea posible
015	Text	Contenido en la ToolBar 003	Texto que muestra el nombre del informe generado



016	AnchorPane	Contiene las tablas que muestran los datos del informe Contenido en el SplitPane 004	Elemento contenedor que alberga las tablas que muestran la respuesta del informe
017	AnchorPane	Contiene la gráfica del informe Contenido en el SplitPane 004	Elemento contenedor que alberga la gráfica que muestra la respuesta del informe
018	TableView	Contenido en el AnchorPane 016	Tabla con una única columna que enumera los parámetros de informe mostrado
019	TableView	Contenido en el AnchorPane 016	Tabla con tantas columnas como horas, días o meses contenga la respuesta y con tantas filas como parámetros contenga el informe
020	StackedBarChart	Contiene los ejes y los datos de la petición Contenido en el AnchorPane 017	Gráfica de barras apiladas con los datos de la respuesta del informe
021	CategoryAxis	Contenido en la StackedBarChart 020	Eje de categorías de la gráfica
022	NumberAxis	Contenido en la StackedBarChart 020	Eje numérico de la gráfica

Tabla 85: Vista Respuesta

### Capa de controlador

La capa de controlador se compone por distintas clases que ayudan a conectar la vista con el modelo. En estas clases como se ha comentado se encuentra toda la lógica necesaria para el correcto funcionamiento, con el objetivo de abstraer totalmente el diseño del modelado y de la interfaz consiguiendo desarrollo paralelos e independientes.

- **Clase Inicial**

La clase Inicial es la clase principal del programa, la que inicia su ejecución y establece toda la interfaz.

Clase Inicial.java			
Atributos	Nombre	Tipo	Descripción
	stagePrincipal	Stage	Atributo que modela la vista al completo, sobre este objeto se construye toda la interfaz
	vistaPrincipal	BorderPane	En este atributo se establece la vista principal en la que se establecen las vistas de

<b>Métodos</b>			bienvenida o de respuesta
	<b>Nombre</b>	<b>Retorno / Argumentos</b>	<b>Descripción</b>
	Inicial	Constructor	Constructor de la clase
	start	void Stage stagePrincipal	Método que inicia la interfaz y establece elementos como el nombre, las proporciones, etc.
	inicializarVistaPrincipal	void	Método que establece la vista inicial que incluye el menú superior y en la que se encuadra el resto de vistas
	mostrarVistaInicial	void	Método que inicializa la vista principal de bienvenida de la aplicación
	mostrarVistaRespuesta	void	Método que inicializa la vista de realización y respuesta de informes de la aplicación
	getStagePrincipal	Stage	Método que devuelve el atributo stagePrincipal
	main	void String[] args	Método principal de la aplicación que la inicializa

Tabla 86: Clase Inicial

- **Clase VistaPrincipalController**

La clase VistaPrincipalController es la clase que controla la vista principal que contiene el menú superior.

**Clase VistaPrincipalController.java**

<b>Atributos</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	inicial	Inicial	Referencia a la clase Inicial
<b>Métodos</b>	<b>Nombre</b>	<b>Retorno / Argumentos</b>	<b>Descripción</b>
	establecerAppInicial	void Inicial inicial	Método que establece el atributo inicial con la referencia a la clase Inicial que recibe
	manejadorRespuesta	void	Método que establece la vista de respuesta cuando se pulsa en el botón del menú

manejadorAcercaDe	void	Método que muestra datos del programa cuando se pulsa en el botón del menú
manejadorSalir	void	Método que cierra el programa cuando se pulsa en el botón del menú

Tabla 87: Clase VistaPrincipalController

- **Clase VistaRespuestaController**

La clase VistaRespuestaController es la clase que controla la vista en la que se realizan peticiones y se recibe la respuesta de cualquier tipo de informe y agregación. Adicionalmente también es esta la vista en la que se pueden exportar los datos consultados.

**Clase VistaRespuestaController.java**

Atributos	Nombre	Tipo	Descripción
	Respuesta	Respuesta[]	Atributo que contiene las respuestas a todos los informes solicitados
	numeroConsultas	int	Atributo en el que se guarda el número de informes que se deben consultar en función de las fechas y el tipo de agregación
	indice	int	Atributo que guarda el índice en el que se encuentra la vista en caso de tratarse de un informe con más de una petición
	fechaActualLD	LocalDate[]	Atributo que modela un array de fechas de los informes generados
	selectorInforme	ChoiceBox<ChoiceBoxUtils>	Atributo que modela el selector del tipo de informe de la petición

selectorAgregacion	ChoiceBox<ChoiceBoxUtils>	Atributo que modela el selector del tipo de agregación de la petición
selectorFechaInicial	DatePicker	Atributo que modela el selector de la fecha inicial de la petición
selectorFechaFinal	DatePicker	Atributo que modela el selector de la fecha final de la petición
botonConsultar	Button	Atributo que modela el botón de consultar de la petición
botonExportar	Button	Atributo que modela el botón de exportar el informe
botonSiguiente	Button	Atributo que modela el botón siguiente de los informes por rango de fechas
fechaActual	TextField	Atributo que modela la fecha del informe mostrado
botonAnterior	Button	Atributo que modela el botón anterior de los informes por rango de fechas
nombreInforme	Text	Atributo que modela el nombre del informe en la vista
tablaParametros	TableView<String>	Atributo que modela la tabla en la que se listan los nombres de los parámetros del informe
tablaContenido	TableView<Double[]>	Atributo que modela la tabla en

<b>Métodos</b>			la que se listan los datos del informe
	grafica	StackedBarChart<String, Number>	Atributo que modela la gráfica en la que se representan los datos del informe
	ejeX	CategoryAxis	Atributo que representa el eje de categorías de la gráfica en la que se representan los datos del informe
	ejeY	NumberAxis	Atributo que representa el eje numérico de la gráfica en la que se representan los datos del informe
	inicial	Inicial	Referencia a la clase inicial
	<b>Nombre</b>	<b>Retorno / Argumentos</b>	<b>Descripción</b>
	VistaRespuestaController	Constructor	Constructor de la clase
	initialize	void	Método que inicializa la vista rellenando desplegables y preparándolo todo para su uso
	establecerApplInicial	void Inicial inicial	Método que establece el atributo inicial con la referencia a la clase Inicial que recibe
	manejadorSeleccionInforme	void	Método que establece el informe de la petición en función de lo escogido en el selector de informe
	manejadorSeleccionAgregacion	void	Método que establece la agregación de la petición en función

		de lo escogido en el selector de agregación
manejadorConsultar	void	Método que realiza la petición en función del tipo de informe, la agregación y las fechas
manejadorExportar	void	Método que llama a exportar
manejadorAnterior	void	Método que muestra el informe anterior
manejadorSiguiente	void	Método que muestra el informe siguiente
getNumeroConsultas	int	Método que devuelve el atributo numeroConsultas
setNumeroConsultas	void int numeroConsultas	Método que establece el atributo numeroConsultas
getIndice	int	Método que devuelve el atributo índice
setIndice	void int indice	Método que establece el atributo indice
mapearNombreInforme	String int informe Int agregacion	Método que establece el texto del nombre del informe en función del tipo y la agregación

Tabla 88: Clase VistaRespuestaController

Como se puede observar, en esta clase se centran todas las acciones principales que puede realizar el usuario con la aplicación, siendo donde se realizan los informes y se visualizan, así como donde se exportan.

Adicionalmente a las clases cuya finalidad exclusiva es la de ejercer de controlador de sus correspondientes vistas, para la lógica de la aplicación se han programado clases auxiliares con el fin de crear un entorno ordenado. Estas clases se definen a continuación.

- **Clase ChoiceBoxUtils**

La clase ChoiceBoxUtils contiene métodos auxiliares referentes a este tipo de objeto, además de contener las constantes que modelan el contenido de los selectores de la aplicación.

Clase ChoiceBoxUtils.java			
Constantes	Nombre	Tipo	Descripción
	PRG_NO_SELECCIONADO	ChoiceBoxUtils	Selector que modela un informe no seleccionado
	DEMANDA_P48	ChoiceBoxUtils	Selector que modela el informe demanda P48
	PROGRAMA_P48	ChoiceBoxUtils	Selector que modela el informe programa P48
	DEMANDA_PBF	ChoiceBoxUtils	Selector que modela el informe demanda PBF
	PROGRAMA_PBF	ChoiceBoxUtils	Selector que modela el informe programa PBF
	DEMANDA_PVP	ChoiceBoxUtils	Selector que modela el informe demanda PVP
	PROGRAMA_PVP	ChoiceBoxUtils	Selector que modela el informe programa PVP
	CONTRATACION_PBF	ChoiceBoxUtils	Selector que modela el informe contratación PBF
	PREV_EOLICA	ChoiceBoxUtils	Selector que modela el informe previsión eólica
	ENERGIA	ChoiceBoxUtils	Selector que modela el informe energía gestionada
	PRECIO	ChoiceBoxUtils	Selector que modela el informe precios marginales
	AGR_NO_SELECCIONADA	ChoiceBoxUtils	Selector que modela una agregación no seleccionada
	AGR_DIARIA	ChoiceBoxUtils	Selector que modela la agregación diaria
	AGR_MENSUAL	ChoiceBoxUtils	Selector que modela la agregación mensual
	AGR_ANUAL	ChoiceBoxUtils	Selector que modela la agregación anual

<b>Atributos</b>	AGR_INTERANUAL	ChoiceBoxUtils	Selector que modela que modela la agregación interanual
	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	id	int	Valor único dentro de un selector para representar la selección
<b>Métodos</b>	valor	String	El nombre de la selección
	<b>Nombre</b>	<b>Retorno / Argumentos</b>	<b>Descripción</b>
	ChoiceBoxUtils	Constructor int id String valor	Constructor completo de la clase
	getId	int	Método que retorna el id de un campo de selección
	toString	String	Método que retorna el valor o nombre de un campo de selección
	rellenarSelectorInformes	ArrayList<ChoiceBoxUtils>	Método que devuelve un array con todos los campos de selección del selector de informes
	rellenarSelectorAgregacion	ArrayList<ChoiceBoxUtils>	Método que devuelve un array con todos los campos de selección del selector de agregación
	validarInforme	boolean int idInforme	Método que valida el informe seleccionado dado un id y devuelve mensaje de error en caso de ser necesario
	validarAgregacion	boolean int idAgregacion	Método que valida la agregación seleccionada dado un id y devuelve mensaje de error en caso de ser necesario

Tabla 89: Clase ChoiceBoxUtils

- **Clase FechaUtils**

La clase FechaUtils contiene métodos auxiliares referentes a la fecha.

<b>Clase</b>	<b>FechaUtils.java</b>		
<b>Métodos</b>	<b>Nombre</b>	<b>Retorno / Argumentos</b>	<b>Descripción</b>
	diaToString	String int dia	Método que retorna el día en formato de dos



		cifras introduciendo un cero delante del número del mes si es necesario
mesToString	String int mes	Método que retorna el mes en formato de dos cifras introduciendo un cero delante del número del mes si es necesario
calcularNumeroDeDias	long LocalDate fechaInicial LocalDate fechaFinal	Método que retorna el número de días comprendido entre dos fechas
calcularNumeroDeMeses	long LocalDate fechaInicial LocalDate fechaFinal	Método que devuelve el número de meses comprendido entre dos fechas
calcularNumeroDeAnyos	long LocalDate fechaInicial LocalDate fechaFinal	Método que devuelve el número de años comprendido entre dos fechas
compararFecha	int LocalDate fechaInicial LocalDate fechaFinal	Método que compara una fecha con otra para saber si es menor, igual o mayor
validarFecha	int LocalDate fechaInicial LocalDate fechaFinal	Método que valida las fechas seleccionadas y devuelve mensaje de error en caso de ser necesario

Tabla 90: Clase FechaUtils

- **Clase GuardarUtils**

La clase FechaUtils contiene métodos auxiliares referentes a la fecha.

**Clase GuardarUtils.java**

<b>Métodos</b>	<b>Nombre</b>	<b>Retorno / Argumentos</b>	<b>Descripción</b>
	guardarArchivo	void Respuesta[] respuesta Stage stage	Método que crea el cuadro de dialogo de guardado y en función de la extensión elegida llama al método necesario
	guardarArchivoXML	boolean Respuesta[] respuesta String rutaArchivo	Método que guarda el informe en extensión XML

guardarArchivoTXTMatlab	boolean Respuesta[] respuesta String rutaArchivo	Método que guarda el informe en extensión TXT adaptado a MatLab
guardarArchivoCSV	boolean Respuesta[] respuesta String rutaArchivo	Método que guarda el informe en extensión CSV

Tabla 91: Clase GuardarUtils

- **Clase ObservableListUtils**

La clase ObservableListUtils contiene métodos auxiliares referentes a este tipo de objetos.

**Clase ObservableListUtils.java**

Métodos	Nombre	Retorno / Argumentos	Descripción
	getNombreDatos	ObservableList<String> String[] nombreDatos	Método que devuelve el nombre de los parámetros en forma de lista observable para poder ser añadidos a la tabla
	getTiempoDatos	ObservableList<String> String[] tiempoDatos	Método que devuelve los tiempos de los datos en forma de lista observable para poder ser añadidos a la tabla
	getDatos	ObservableList<Double[]> Double[][] datos	Método que devuelve los datos de la respuesta en forma de lista observable para poder ser añadidos a la tabla

Tabla 92: Clase ObservableListUtils

- **Clase PeticionUtils**

La clase PeticionUtils contiene métodos auxiliares referentes a este tipo de objetos. En esta clase se realiza la petición al servidor y se transforma en un objeto Respuesta válido, para poder manejar los informes durante todo el flujo de eventos, desde su visualización hasta su guardado.

**Clase PeticionUtils.java**

Constantes	Nombre	Tipo	Descripción
	USER_AGENT	String	Constante que indica el agente de usuario para la petición
Métodos	Nombre	Retorno / Argumentos	Descripción
	realizarPeticion	Respuesta LocalDate fechaInicial int tipoPeticion	Método que recibe los datos necesarios para realizar la petición al

	int tipoAgregacion	servidor de Red Eléctrica y responde un objeto Respuesta con el informe recibido adaptado al modelo
mapearTipoPeticion	String int tipoPeticion	Método auxiliar para mapear acorde a lo necesario el formato del tipo de petición en función del id interno recibido
mapearFechaPeticion	String LocalDate fecha int tipoAgregacion	Método auxiliar para mapear acorde a lo necesario el formato de la fecha
mapearAgregacionPeticion	String int tipoAgregacion	Método auxiliar para mapear acorde a lo necesario el formato de la agregación en función del id interno recibido

Tabla 93: Clase PeticionUtils

- **Clase RespuestaParser**

La clase RespuestaParser es el corazón del programa. Si bien la petición es importante (sin ella no tendríamos datos), la parte más relevante y compleja del programa se realiza en esta clase, manejando los datos que se reciben para estructurarlos en forma de objeto Respuesta, que como se ha definido anteriormente, se trata de un objeto preparado para poder contener cualquier tipo de informe de todos los modelados. El trabajo se realiza con la ayuda de una librería externa llamada jsoup, un parser HTML programado en Java del que se hablará en apartados posteriores. A pesar de su complejidad, la clase sólo contiene un método que realiza todas las operaciones.

Clase RespuestaParser.java			
Métodos	Nombre	Retorno / Argumentos	Descripción
	parsearRespuesta	Respuesta String html LocalDate fecha int tipoPeticion int tipoAgregacion	Método

Tabla 94: Clase RespuestaParser

- **Clase StackedBarChartUtils**

La clase StackedBarChartUtils contiene el método encargado de crear la gráfica con los datos de la respuesta.

Clase <b>StackedBarChartUtils.java</b>			
Métodos	Nombre	Retorno / Argumentos	Descripción
	rellenarGrafica	StackedBarChart<String, Number> CategoryAxis ejeX NumberAxis ejeY Respuesta respuesta	Método que dados los datos de la respuesta devuelve una gráfica de barras apiladas

Tabla 95: Clase StackedBarChartUtils

- **Clase TableViewUtils**

La clase TableViewUtils contiene los métodos encargados de crear la tabla con los datos de la respuesta, además de los métodos auxiliares necesarios.

Clase <b>TableViewUtils.java</b>			
Métodos	Nombre	Retorno / Argumentos	Descripción
	rellenarTablaParametros	TableView<String> Respuesta respuesta	Método que dados los datos de la respuesta devuelve la tabla con los parámetros rellena
	rellenarTablaDatos	TableView<Double[]> Respuesta respuesta	Método que dados los datos de la respuesta devuelve la tabla con los datos rellena
	encontrarScrollBarVertical	ScrollBar TableView<?> tabla	Método que devuelve la propiedad de la tabla del desplazamiento vertical con el objetivo de poder sincronizar el desplazamiento de tablas

Tabla 96: Clase TableViewUtils

- **Clase Utils**

La clase Utils contiene métodos auxiliares genéricos para el correcto funcionamiento de la herramienta.

Clase <b>Utils.java</b>			
Métodos	Nombre	Retorno / Argumentos	Descripción
	devolverDoubleValido	Double String numero	Dada una cadena de texto, este método devuelve un Double con su valor o un null si no es válido
	devolverDoubleCero	Double Double numero	Dado un double, este método devuelve un

		Double con su valor o un 0 si no es válido
devolverStringformatoDouble	String String numero	Método que formatea los double a formato estándar para su correcto uso interno
formatearDouble	String Double numero	Método que formatea los Double a formato español para su visualización
transponerMatriz	Double[][] Double[][] matriz	Método que transpone la matriz de datos
eliminarNulls	Double[][] Double[][] matriz	Método que elimina los nulls que pueda contener la matriz de datos y los convierte en cero

Tabla 97: Clase Utils

### *Diseño de la interfaz*

La interfaz basada en la librería propietaria de Oracle JavaFX, se compone de varias perspectivas que a su vez se componen de vistas. En esta sección se explican estos conceptos además de explicar las perspectivas que forman parte de la versión final del programa. Los prototipos anteriores a la fase final de desarrollo de la interfaz se encuentran en el anexo adjunto en este documento [Anexo V. Prototipo

Una perspectiva es cada una de las pantallas que componen la aplicación y que se diferencian totalmente por su función, es decir, cada una de las perspectivas modela una sección de la aplicación con objetivos completamente distintos, por poner un ejemplo, en un programa de edición de texto como Microsoft Word, al abrir el programa encontraríamos una perspectiva en la que se nos invita a crear o abrir un archivo existente, y al modificar un archivo nos encontraríamos en la perspectiva de edición. Las perspectivas se componen a su vez de vistas. Siguiendo el mismo ejemplo, en el programa de edición de texto, una vista sería por ejemplo el cuadro de dialogo lateral de búsqueda y otra el área de edición de texto. La interfaz que se ha diseñado, se compone de dos perspectivas; la de bienvenida y la de respuesta.

### *Perspectiva de bienvenida*

En la perspectiva de bienvenida se muestra un diagrama del funcionamiento de la aplicación además del menú de la misma y un breve texto explicativo de cómo comenzar a realizar informes. En la figura 20 se aprecia esta perspectiva dividida en vistas para su posterior explicación detallada.

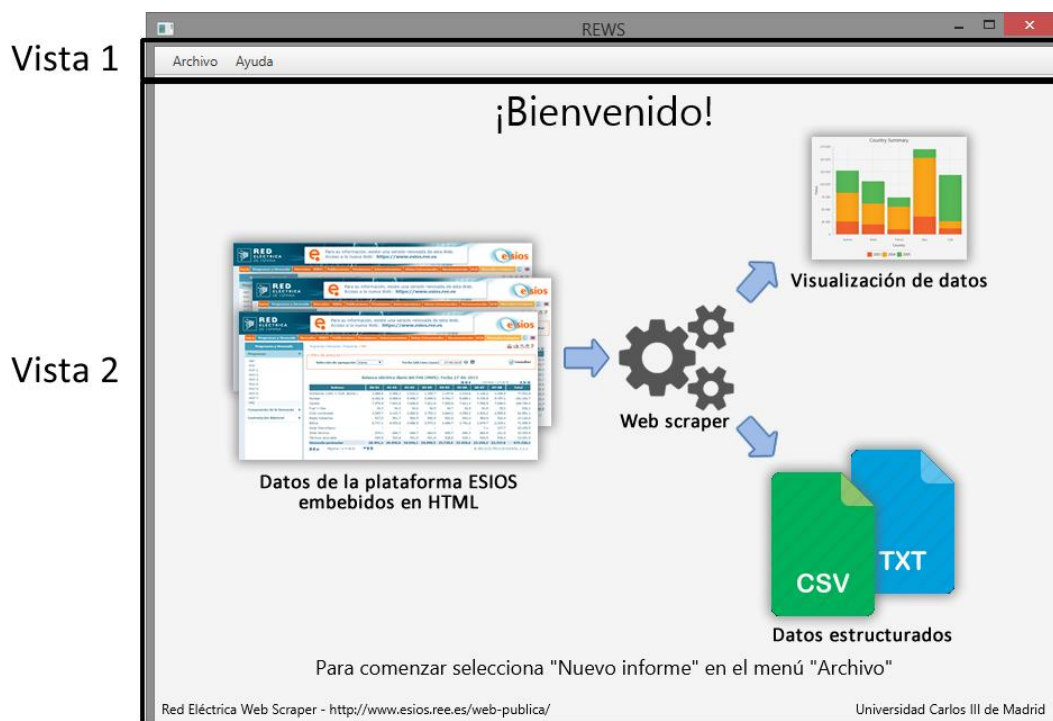


Figura 20: Perspectiva de bienvenida

Se descompone en dos vistas; la vista de menú y la vista de imagen como se puede observar en la figura superior. El menú de la aplicación (Vista 1) es común a todas las perspectivas y da la opción de navegar hasta la perspectiva de informes, observar información de la aplicación en forma de ventana emergente y salir de la misma. La vista de imagen (Vista 2) muestra en un diagrama el funcionamiento interno de la aplicación de manera sencilla, además de contar con un texto explicativo para ayudar al usuario a empezar a crear los informes de datos de Red Eléctrica.

### *Perspectiva de respuesta*

La perspectiva de respuesta es donde el usuario realizará las acciones principales que le permite el programa y no únicamente la de observar la respuesta de los informes. Es donde se rellenan los datos de la petición del informe, donde se muestran los datos de la respuesta del mismo y desde donde se pueden exportar. En la figura 21 se observa la perspectiva de respuesta dividida en las diferentes vistas que la forman para poder ser explicada con claridad.

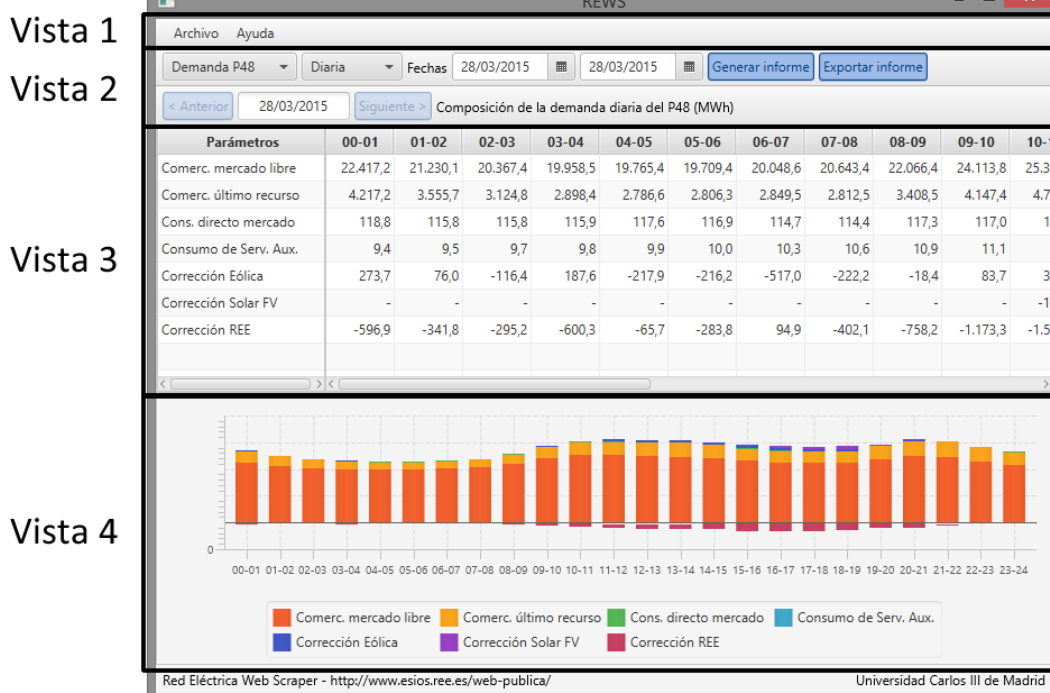


Figura 21: Perspectiva de respuesta

En este caso la perspectiva se divide en cuatro vistas. La primera, al igual que en la perspectiva de bienvenida se trata del menú de la aplicación (Vista 1). En este caso, la segunda vista se trata del menú para realizar la petición (Vista 2). La tercera y la cuarta vista es donde se muestran los datos de la respuesta. La tercera es una tabla en la que se muestra cada parámetro en forma de fila y su extensión depende del tipo de petición realizada (Vista 3). La cuarta y última se trata de una gráfica en la que se representan los datos de la respuesta en forma de barras apiladas (Vista 4).

## Implementación

En este apartado se describe la fase de implementación además de todas las herramientas usadas para la misma. Adicionalmente se explican partes del proyecto que ayudan a establecer una visión global, como su organización interna.

### Entorno de desarrollo

Para el desarrollo de una aplicación elaborada, no es imprescindible pero si deseable el uso de una herramienta de desarrollo adaptada al lenguaje que se va a usar. Dado que el desarrollo de la aplicación completa se realiza en java, se ha usado para su implementación el entorno de desarrollo por excelencia para este lenguaje, es decir, Eclipse.

Eclipse es un IDE (Integrated Development Environment) o entorno de desarrollo integrado de software libre desarrollado por la Fundación Eclipse. Para el desarrollo de este proyecto se ha usado la versión 4.4.1 Luna [ECLIPSE]. La decisión del uso de este entorno sobre otros como por ejemplo IntelliJ IDEA o NetBeans es en el primer caso el hecho de que Eclipse sea software libre y en el segundo el hecho de que durante toda la carrera y en el entorno laboral se haya usado Eclipse sobre NetBeans. En la figura 22 se muestra la distribución del entorno de desarrollo durante esta misma fase.

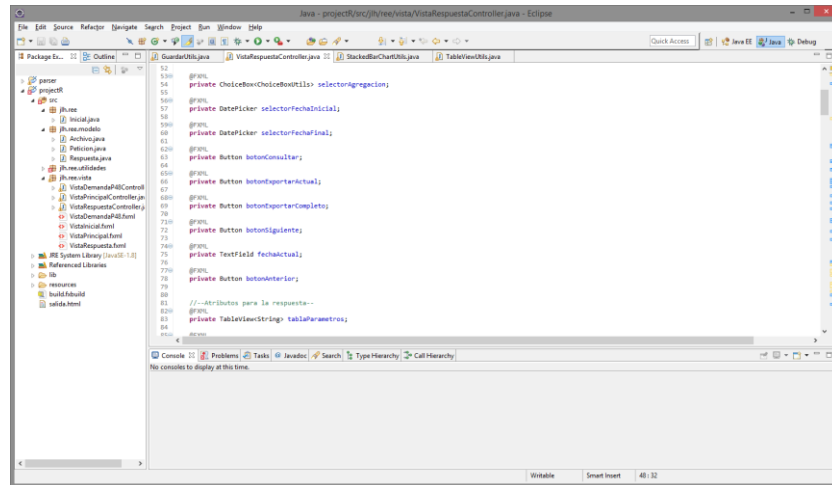


Figura 22: Entorno de desarrollo Eclipse con el proyecto abierto

Para el desarrollo de la interfaz, Oracle proporciona una herramienta visual con la que construir las interfaces en JavaFX de forma cómoda. Dicha herramienta se llama JavaFX Scene Builder y ha sido usada en su versión 2.0. En la figura 23 se observa el constructor de interfaces con una interfaz del proyecto abierta.

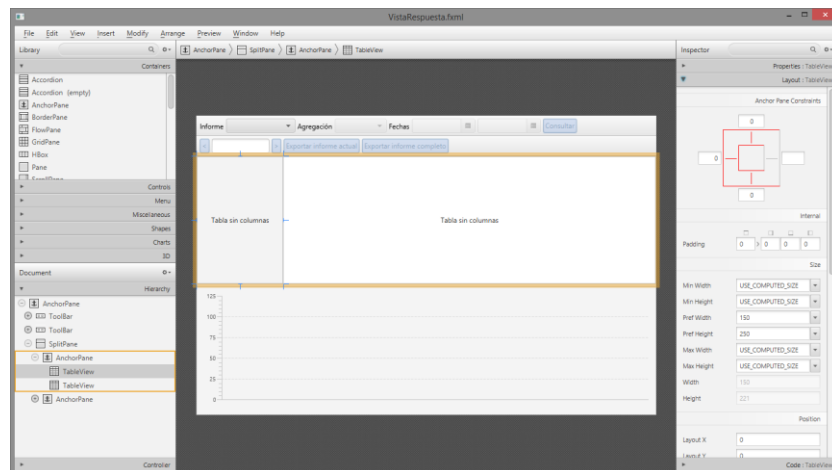


Figura 23: JavaFX Scene Builder con un archivo del proyecto abierto

Si bien esta herramienta ayuda a crear las interfaces de forma sencilla, parte del trabajo se debe hacer a mano codificando. Para esto se usó también Eclipse ya que está bien integrado con los proyectos de tipo JavaFX.

### Organización del proyecto

En este apartado se define la organización interna del proyecto en la fase de implementación. Antes de nada, y para comenzar, se inicia un proyecto desde cero en el IDE Eclipse. La peculiaridad de este proyecto es que cuenta con interfaces de JavaFX por lo que a la hora de comenzar el



proyecto hay que indicarlo, es decir, no vale con crear un proyecto normal de Java, se debe crear un proyecto de JavaFX tal y como se ve en la figura 24.

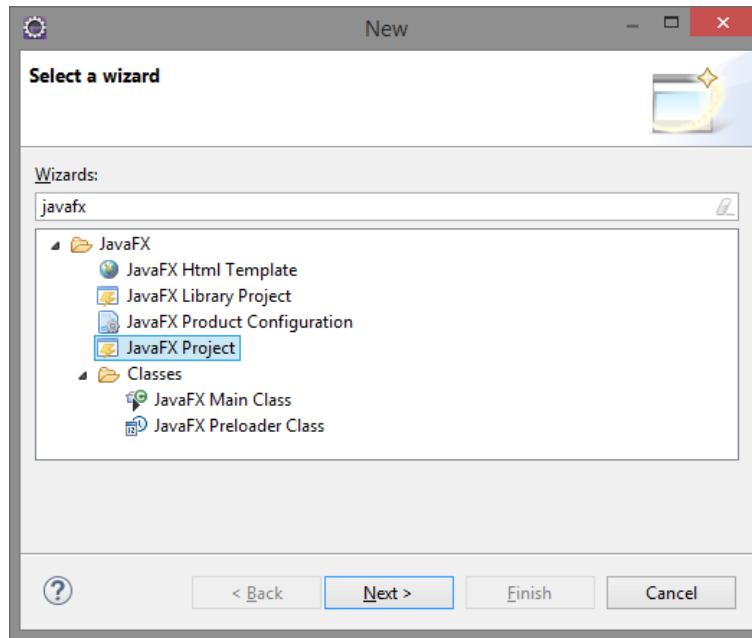


Figura 24: Asistente de creación de proyectos de JavaFX en Eclipse

Una vez dado el primer paso, se debe establecer la organización primordial del proyecto a nivel interno. Puesto que estamos usando una librería basada en MVC o Modelo Vista Controlador, lo más lógico es establecer una ordenación de las clases que componen el programa en función de este tipo de programación de interfaces. De esta forma, establecemos distintos paquetes en los que se distribuirán en función de a que parte pertenezcan, siendo estas una genérica, otra para el modelo y otra para la vista y los controladores de las vistas. En la figura 25 se aprecia la organización de los paquetes. El nombre de los paquetes comunes viene dado por las siglas de Jorge López Hornero y Red Eléctrica Española.

```
jlh.ree
jlh.ree.modelo
jlh.ree.utilidades
jlh.ree.vista
```

Figura 25: Paquetes que componen el proyecto

Una de las restricciones de JavaFX afecta a la forma en la que deben nombrarse y posicionarse los controladores de las vistas, teniendo que estar en el mismo paquete y llamándose igual que el archivo de la vista añadiendo “Controller” al final del nombre. Es por esto que no existen dos paquetes distintos para las vistas (archivos FXML) y los controladores (archivos Java). En la figura 26 se aprecia un ejemplo de esto mismo.

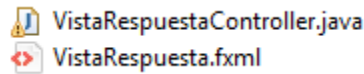


Figura 26: Un archivo de vista y su respectiva clase controladora

Adicionalmente a las clases programadas para la herramienta, se han usado librerías para facilitar ciertos aspectos de la programación. La primera de las dos librerías es jsoup, librería encargada de obtener la respuesta del servidor en HTML y convertirla en un objeto de Java, aportando versatilidad y facilidad de manejo sobre la respuesta. Se trata de una librería de código abierto distribuida bajo una licencia del MIT (Massachusetts Institute of Technology) [JSOUP]. La versión que se ha utilizado ha sido la 1.8.1.

La segunda librería afecta a la parte gráfica del programa. Se trata de ControlsFX y sirve de ayuda a la hora de crear diálogos de error o información dentro de las interfaces de JavaFX. En la figura 27 se aprecia un ejemplo de diálogo de error de la aplicación creado con esta librería. Se distribuye desde su página web con una licencia gratuita que únicamente obliga a mencionar su uso dentro de la documentación del software que la contenga [CONTROLSFX]. La versión usada ha sido la 8.20.8.

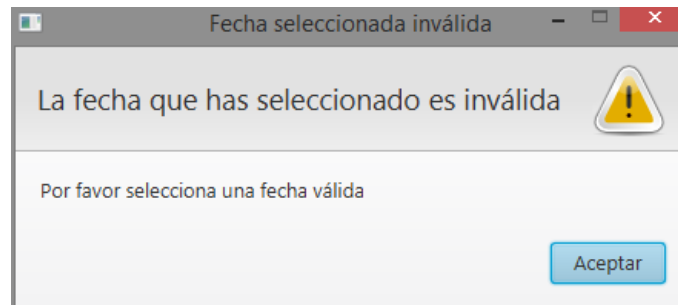


Figura 27: Diálogo de error creado con ControlsFX

La estructura de archivos del proyecto sigue el esquema normal establecido por defecto en los proyectos de Eclipse como queda expuesto en la figura 28. En ella se encuentra una carpeta donde se guardan las preferencias del proyecto para su apariencia dentro del IDE, una carpeta con los binarios (las clases de java compiladas y listas para su ejecución), una carpeta con las librerías importadas mencionadas anteriormente y una carpeta con los archivos Java antes de su compilación, siendo esta carpeta donde se guarda el código con la estructura definida por los paquetes del proyecto. El resto de archivos son archivos que Eclipse necesita para la correcta apertura del proyecto y para la ejecución del programa.

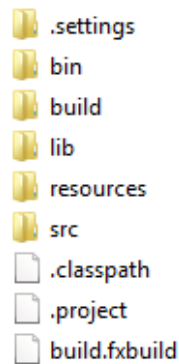


Figura 28: Estructura de carpetas del proyecto

En la figura superior se puede observar dos carpetas no definidas anteriormente. La carpeta “resources” se escapa de la estructura general de los proyectos de Eclipse y ha sido creada a mano intentando seguir una estructura lógica. En ella se guardan los archivos externos necesarios para el correcto funcionamiento y visualización del programa como pueden ser las imágenes o las hojas de estilo. En la figura 29 se puede ver el contenido de la misma.



Figura 29: Contenido de la carpeta resources

Adicionalmente, la carpeta “build” se genera automáticamente al crear el ejecutable del proyecto. En su interior se genera una estructura de carpetas en las que se deben meter manualmente los recursos de la carpeta resources y en la que se debe cambiar cierta información como el versionado, los distintos iconos de la aplicación y parámetros para la correcta realización de los archivos ejecutables ya sea el genérico .jar o los puntuales para cada sistema operativo. En la figura 30 se puede ver el interior de la carpeta tras generar un ejecutable de prueba.

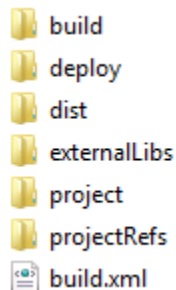


Figura 30: Contenido de la carpeta build

El archivo “build.xml” es un script que se genera de forma automática y cuya misión es compilar el proyecto y realizar los ejecutables. En él se encuentra toda la información necesaria para que el proyecto funcione además de poder modificarse ciertos aspectos como la ruta de los archivos de recursos, la versión del programa, la ruta de los iconos, etc. Este archivo debe ser ejecutado

mediante Apache Ant [ANT] para conseguir los ejecutables necesarios para usar el programa. Apache Ant es un compilador construido en Java tiene la ventaja de ser compatible con cualquier sistema operativo en el que Java sea ejecutable, en lugar de depender de las ordenes de la consola de cada sistema.

Las interfaces creadas en JavaFX permiten la modificación de los componentes visuales que la forman mediante hojas de estilo CSS modificadas ligeramente con sintaxis propia de JavaFX. Estas hojas de estilo se guardan dentro de la carpeta “style” reflejada en la figura superior. En el ejemplo mostrado en la figura 31 se ven los componentes necesarios para colorear y redondear un botón de la interfaz junto a su resultado.

```
.button{
    -fx-text-fill: #004080;
    -fx-border-color: #004080;
    -fx-background-color: #9fbce8;
    -fx-border-radius: 3;
    -fx-padding: 3;
}
```

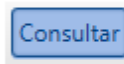


Figura 31: Botón de la interfaz modificado con hojas de estilo

Los archivos que componen la interfaz se guardan junto a las clases de Java antes de ser compiladas en la carpeta src. En esta carpeta como ya se ha comentado, se refleja la jerarquía de carpetas definida por los paquetes del proyecto, por lo que las vistas se encontrarán dentro de la ruta src/jlh/ree/vista. Los archivos FXML (archivos de interfaz) se constituyen con una sintaxis de XML modificada con elementos propios de JavaFX. En la figura 32 se puede apreciar un ejemplo del contenido de un archivo FXML y su resultado en tiempo de ejecución.

```
<?xml version="1.0" encoding="UTF-8"?>
<?import javafx.scene.control.*?>
<?import java.lang.*?>
<?import javafx.scene.layout.*?>

<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" pre ...
  <children>
    <TabPane prefHeight="200.0" prefWidth="200.0" tabClosingPolicy="UNAVAILABLE">
      <tabs>
        <Tab text="Pestaña 1">
          <content>
            <AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="180.0" prefWidth="200.0">
              <children>
                <Button layoutX="74.0" layoutY="73.0" mnemonicParsing="false" text="Prueba" />
              </children>
            </AnchorPane>
          </content>
        </Tab>
        <Tab text="Pestaña 2">
          <content>
            <AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="180.0" prefWidth="200.0" />
          </content>
        </Tab>
      </tabs>
    </TabPane>
  </children>
</AnchorPane>
```



Figura 32: Ejemplo de archivo FXML y su representación

El programa permite al usuario la exportación de informes en distintos formatos usando los cuadros de diálogo del propio sistema operativo en el que se está ejecutando, de forma que resulte totalmente intuitivo. En la figura 33 se puede apreciar una muestra de un informe exportado en formato compatible con MatLab.

UNIVERSIDAD CARLOS III DE MADRID

```
% DEM_P48 - 2015-03-30 - 2015-08-17
% Comerc. mercado libre - Comerc. último recurso - ...
datos = [19919.3 3765.7 115.9 8.3 293.5 0.0 -743.5
19139.1 3195.2 116.2 8.2 482.5 0.0 -974.8
18638.0 2862.3 115.2 8.1 1190.9 0.0 -1316.2
18491.3 2670.2 115.3 8.0 932.7 0.0 -1148.2
18565.1 2585.8 115.6 8.0 1256.0 0.0 -1668.8
19324.9 2626.4 117.5 8.0 1308.4 0.0 -1763.7
21336.9 2905.7 117.1 8.1 753.7 0.0 -1898.2
23888.7 3500.9 118.3 8.1 361.9 0.0 -813.9
25870.0 4020.4 119.6 8.5 276.0 -200.0 -1354.7
27245.3 4102.0 118.7 9.0 488.1 -200.0 -1328.5
28047.6 4319.4 118.9 9.5 197.3 -300.0 -1045.1
28029.5 4330.8 125.9 9.5 293.5 -400.0 -593.5
27795.0 4342.8 126.0 9.4 599.6 -300.0 -797.9
27342.2 4587.5 122.6 9.2 602.3 -200.0 -954.8
26531.6 4360.3 124.8 8.9 202.6 -200.0 -509.3
...]
```

Figura 33: Ejemplo de informe exportado compatible con MatLab

## 5 Evaluación

Todo proceso de desarrollo de software tiene como fase previa a su uso por parte de los usuarios finales la fase de evaluación. En esta fase se pretende comprobar si el resultado obtenido después del desarrollo satisface las necesidades establecidas durante el inicio del proyecto. Para comprobarlo, se han establecido distintos casos de prueba que evalúan el comportamiento de la aplicación en situaciones concretas. Adicionalmente a los casos de prueba, se ha realizado un cuestionario de usabilidad a un grupo cerrado de diez usuarios a fin de determinar su facilidad de uso y aprendizaje.

Los resultados de ambos procesos de evaluación se muestran en los siguientes apartados, junto al análisis de los resultados obtenidos.

### 5.1 Proceso de evaluación

Existen dos tipos bien diferenciados de propósitos de evaluación; la verificación y la validación. El primer tipo busca comprobar si el resultado final del programa satisface los requisitos establecidos durante el análisis del sistema mientras que el segundo tipo busca comprobar si el producto final hace lo que el usuario espera. La verificación en este caso se realizará mediante un conjunto de casos de prueba mientras que la validación se realizará mediante un cuestionario de usabilidad realizado a un grupo de usuarios.

En los siguientes apartados se describirán los procesos de verificación y validación en detalle con el fin de realizar la evaluación del sistema.

#### Casos de prueba

Los casos de uso descritos, deben ser verificados mediante casos de prueba. Para la correcta definición de los casos de prueba se debe seguir una plantilla para clasificarlos a todos por igual. Esta plantilla correspondiente a los casos de prueba se muestra a continuación en la tabla 98:

ID	
Nombre	
Caso de uso	
Descripción	
Precondiciones	
Flujo de eventos	
Verificación	

Tabla 98: Plantilla para los casos de prueba

A continuación se explican todos los campos de los que se compone la plantilla de los casos de prueba:

- **ID:** los casos de prueba se distinguen mediante un identificador único que no puede ser repetido. Este identificador se forma empezando con las siglas de caso de prueba (CP), seguido de un guion y un código numérico de tres cifras que se incrementaría por unidades para cada uno de los casos de prueba de forma ordenada. El primero de los casos de prueba recibirá el ID CP-001.
- **Nombre:** frase breve descriptiva sobre el caso de prueba.
- **Caso de uso:** caso o casos de uso relacionados con el caso de prueba en cuestión.

- **Descripción:** en este campo se describe brevemente el caso de prueba en forma de pequeño párrafo sin restricciones concretas más que la extensión razonable.
- **Precondiciones:** en las precondiciones se describen las condiciones necesarias para poder realizar el caso de prueba.
- **Flujo de eventos:** aquí se describe paso por paso las acciones que se deben realizar para verificar el caso de prueba.
- **Verificación:** en este campo se describe la que debe ser la respuesta correcta del sistema frente al flujo de eventos propuesto.

Los casos de prueba del sistema se exponen usando la plantilla que ha sido descrita. Para evitar redundancias innecesarias, se ha decidido crear casos de prueba genéricas en lugar de crear individuales para cada tipo de informe y agregación.

ID CP-001	
<b>Nombre</b>	Realizar un informe cualquiera
<b>Caso de uso</b>	CU-001, CU-003, CU-004
<b>Descripción</b>	El usuario desea realizar un informe de cualquier tipo y agregación
<b>Precondiciones</b>	El usuario debe contar con la máquina virtual de Java instalada y el programa en su equipo además de contar con conexión a internet
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El usuario abre el programa</li> <li>2. El usuario pulsa en el menú “Nuevo informe” dentro del apartado “Archivo”</li> <li>3. El usuario selecciona el tipo de informe deseado</li> <li>4. El usuario selecciona el tipo de agregación deseada</li> <li>5. El usuario selecciona la fecha deseada para el informe</li> <li>6. El usuario pulsa el botón “Consultar”</li> <li>7. El usuario repite el proceso con todas las combinaciones de informe y agregación</li> </ol>
<b>Verificación</b>	<ul style="list-style-type: none"> <li>• El usuario es capaz de abrir el programa</li> <li>• El usuario es capaz de llegar a la pantalla de informes</li> <li>• El usuario es capaz de seleccionar el informe deseado</li> <li>• El usuario es capaz de seleccionar el tipo de agregación deseada</li> <li>• El usuario es capaz de seleccionar la fecha deseada</li> <li>• El usuario es capaz de realizar la consulta</li> <li>• El informe se muestra correctamente independientemente de las opciones escogidas</li> </ul>

Tabla 99: Caso de prueba CP-001

ID CP-002	
<b>Nombre</b>	Realizar un informe cualquiera por rangos de fechas
<b>Caso de uso</b>	CU-002, CU-003, CU-004
<b>Descripción</b>	El usuario desea realizar un informe de cualquier tipo y agregación y con un rango de fechas concreto
<b>Precondiciones</b>	El usuario debe contar con la máquina virtual de Java instalada y el programa en su equipo además de contar con conexión a internet
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El usuario abre el programa</li> </ol>

	<ol style="list-style-type: none"> <li>El usuario pulsa en el menú “Nuevo informe” dentro del apartado “Archivo”</li> <li>El usuario selecciona el tipo de informe deseado</li> <li>El usuario selecciona el tipo de agregación deseada</li> <li>El usuario selecciona las fechas deseadas para el informe acotando fecha inicial y final</li> <li>El usuario pulsa el botón “Consultar”</li> <li>El usuario repite el proceso con todas las combinaciones de informe y agregación</li> </ol>
<b>Verificación</b>	<ul style="list-style-type: none"> <li>El usuario es capaz de abrir el programa</li> <li>El usuario es capaz de llegar a la pantalla de informes</li> <li>El usuario es capaz de seleccionar el informe deseado</li> <li>El usuario es capaz de seleccionar el tipo de agregación deseada</li> <li>El usuario es capaz de seleccionar las fechas deseadas</li> <li>El usuario es capaz de realizar la consulta</li> <li>El primer informe se muestra correctamente independientemente de las opciones escogidas</li> </ul>

Tabla 100: Caso de prueba CP-002

<b>ID CP-003</b>	
<b>Nombre</b>	Cambiar de informe
<b>Caso de uso</b>	CU-002, CU-003, CU-004
<b>Descripción</b>	El usuario desea cambiar de informe después de realizar un informe por rangos de fechas
<b>Precondiciones</b>	<p>El usuario debe contar con la máquina virtual de Java instalada y el programa en su equipo además de contar con conexión a internet</p> <p>o</p> <p>El usuario debe haber realizado un informe de rangos de fechas</p>
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>El usuario abre el programa</li> <li>El usuario pulsa en el menú “Nuevo informe” dentro del apartado “Archivo”</li> <li>El usuario selecciona el tipo de informe deseado</li> <li>El usuario selecciona el tipo de agregación deseada</li> <li>El usuario selecciona las fechas deseadas para el informe acotando fecha inicial y final</li> <li>El usuario pulsa el botón “Consultar”</li> <li>El usuario repite el proceso con todas las combinaciones de informe y agregación</li> <li>El usuario cambia de informe entre los del rango de fechas</li> </ol> <p>o</p> <ol style="list-style-type: none"> <li>El usuario ha realizado un informe por rangos de fechas con anterioridad</li> <li>El usuario cambia de informe entre los del rango de fechas</li> </ol>
<b>Verificación</b>	<ul style="list-style-type: none"> <li>El usuario es capaz de cambiar entre informes después de realizar un informe por rangos de fechas</li> </ul>

Tabla 101: Caso de prueba CP-003



<b>ID CP-004</b>	
<b>Nombre</b>	Realizar un informe cualquiera con agregación diaria en las fechas de cambio de hora
<b>Caso de uso</b>	CU-001, CU-003, CU-004
<b>Descripción</b>	El usuario desea realizar un informe de cualquier tipo y agregación diaria en los días de cambio de fecha
<b>Precondiciones</b>	El usuario debe contar con la máquina virtual de Java instalada y el programa en su equipo además de contar con conexión a internet
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El usuario abre el programa</li> <li>2. El usuario pulsa en el menú “Nuevo informe” dentro del apartado “Archivo”</li> <li>3. El usuario selecciona cualquier tipo de informe</li> <li>4. El usuario selecciona agregación diaria</li> <li>5. El usuario selecciona uno de los días de cambio de hora del último año</li> <li>6. El usuario pulsa el botón “Consultar”</li> <li>7. El usuario repite el proceso con el otro día de cambio de hora</li> </ol>
<b>Verificación</b>	<ul style="list-style-type: none"> <li>• El usuario es capaz de abrir el programa</li> <li>• El usuario es capaz de llegar a la pantalla de informes</li> <li>• El usuario es capaz de seleccionar un informe de la lista</li> <li>• El usuario es capaz de seleccionar el tipo de agregación diaria</li> <li>• El usuario es capaz de seleccionar las fechas deseadas</li> <li>• El usuario es capaz de realizar la consulta</li> <li>• El informe se muestra correctamente adaptado a los días de 23h y 25h</li> </ul>

Tabla 102: Caso de prueba CP-004

<b>ID CP-005</b>	
<b>Nombre</b>	Realizar un informe cualquiera con agregación mensual seleccionando el mes en curso
<b>Caso de uso</b>	CU-001, CU-003, CU-004
<b>Descripción</b>	El usuario desea realizar un informe de cualquier tipo y agregación mensual en el mes en curso
<b>Precondiciones</b>	El usuario debe contar con la máquina virtual de Java instalada y el programa en su equipo además de contar con conexión a internet
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El usuario abre el programa</li> <li>2. El usuario pulsa en el menú “Nuevo informe” dentro del apartado “Archivo”</li> <li>3. El usuario selecciona cualquier tipo de informe</li> <li>4. El usuario selecciona agregación mensual</li> <li>5. El usuario selecciona el mes en curso</li> <li>6. El usuario pulsa el botón “Consultar”</li> </ol>
<b>Verificación</b>	<ul style="list-style-type: none"> <li>• El usuario es capaz de abrir el programa</li> <li>• El usuario es capaz de llegar a la pantalla de informes</li> <li>• El usuario es capaz de seleccionar un informe de la lista</li> </ul>

## UNIVERSIDAD CARLOS III DE MADRID

	<ul style="list-style-type: none"> <li>• El usuario es capaz de seleccionar el tipo de agregación mensual</li> <li>• El usuario es capaz de seleccionar el mes en curso</li> <li>• El usuario es capaz de realizar la consulta</li> <li>• El informe se muestra correctamente adaptado a los días transcurridos del mes en curso</li> </ul>
--	---

Tabla 103: Caso de prueba CP-005

<b>ID CP-006</b>	
<b>Nombre</b>	Realizar un informe cualquiera con agregación anual seleccionando el año en curso
<b>Caso de uso</b>	CU-001, CU-003, CU-004
<b>Descripción</b>	El usuario desea realizar un informe de cualquier tipo y agregación anual en el año en curso
<b>Precondiciones</b>	El usuario debe contar con la máquina virtual de Java instalada y el programa en su equipo además de contar con conexión a internet
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El usuario abre el programa</li> <li>2. El usuario pulsa en el menú “Nuevo informe” dentro del apartado “Archivo”</li> <li>3. El usuario selecciona cualquier tipo de informe</li> <li>4. El usuario selecciona agregación anual</li> <li>5. El usuario selecciona el año en curso</li> <li>6. El usuario pulsa el botón “Consultar”</li> </ol>
<b>Verificación</b>	<ul style="list-style-type: none"> <li>• El usuario es capaz de abrir el programa</li> <li>• El usuario es capaz de llegar a la pantalla de informes</li> <li>• El usuario es capaz de seleccionar un informe de la lista</li> <li>• El usuario es capaz de seleccionar el tipo de agregación anual</li> <li>• El usuario es capaz de seleccionar el año en curso</li> <li>• El usuario es capaz de realizar la consulta</li> <li>• El informe se muestra correctamente adaptado a los meses transcurridos del año en curso</li> </ul>

Tabla 104: Caso de prueba CP-006

<b>ID CP-007</b>	
<b>Nombre</b>	Guardar un informe cualquiera normal
<b>Caso de uso</b>	CU-001, CU-005
<b>Descripción</b>	El usuario desea realizar un informe de cualquier tipo y agregación y guardarlo
<b>Precondiciones</b>	<p>El usuario debe contar con la máquina virtual de Java instalada y el programa en su equipo además de contar con conexión a internet</p> <p style="text-align: center;">o</p> <p>El usuario debe haber realizado un informe</p>
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El usuario abre el programa</li> <li>2. El usuario pulsa en el menú “Nuevo informe” dentro del apartado “Archivo”</li> <li>3. El usuario selecciona el tipo de informe deseado</li> </ol>

	<ol style="list-style-type: none"> <li>El usuario selecciona el tipo de agregación deseada</li> <li>El usuario selecciona la fecha deseada para el informe</li> <li>El usuario pulsa el botón “Consultar”</li> <li>El usuario exporta el informe en cualquier extensión</li> <li>El usuario repite el proceso de exportación con todas las extensiones</li> <li>El usuario repite el proceso con todas las combinaciones de informe y agregación</li> </ol> <p style="text-align: center;">o</p> <ol style="list-style-type: none"> <li>El usuario ha realizado un informe con anterioridad</li> <li>El usuario exporta el informe en cualquier extensión</li> <li>El usuario repite el proceso de exportación con todas las extensiones</li> </ol>
<b>Verificación</b>	<ul style="list-style-type: none"> <li>El informe se exporta correctamente independientemente de la extensión escogida</li> </ul>

Tabla 105: Caso de prueba CP-007

<b>ID CP-008</b>	
<b>Nombre</b>	Guardar un informe cualquiera por rango de fechas
<b>Caso de uso</b>	CU-002, CU-005
<b>Descripción</b>	El usuario desea realizar un informe de cualquier tipo y agregación por rango de fechas y guardarlo
<b>Precondiciones</b>	<p>El usuario debe contar con la máquina virtual de Java instalada y el programa en su equipo además de contar con conexión a internet</p> <p style="text-align: center;">o</p> <p>El usuario debe haber realizado un informe por rango de fechas</p>
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>El usuario abre el programa</li> <li>El usuario pulsa en el menú “Nuevo informe” dentro del apartado “Archivo”</li> <li>El usuario selecciona el tipo de informe deseado</li> <li>El usuario selecciona el tipo de agregación deseada</li> <li>El usuario selecciona las fechas deseadas para el informe acotando fecha inicial y final</li> <li>El usuario pulsa el botón “Consultar”</li> <li>El usuario repite el proceso con todas las combinaciones de informe y agregación</li> <li>El usuario exporta el informe en cualquier extensión</li> <li>El usuario repite el proceso de exportación con todas las extensiones</li> <li>El usuario repite el proceso con todas las combinaciones de informe y agregación</li> </ol> <p style="text-align: center;">o</p> <ol style="list-style-type: none"> <li>El usuario ha realizado un informe por rango de fechas con anterioridad</li> <li>El usuario exporta el informe en cualquier extensión</li> <li>El usuario repite el proceso de exportación con todas las extensiones</li> </ol>
<b>Verificación</b>	<ul style="list-style-type: none"> <li>El informe se exporta correctamente independientemente de la extensión escogida</li> </ul>

Tabla 106: Caso de prueba CP-008

<b>ID CP-009</b>	
<b>Nombre</b>	Ver la información del programa
<b>Caso de uso</b>	CU-006
<b>Descripción</b>	El usuario desea ver la información del programa
<b>Precondiciones</b>	El usuario debe contar con la máquina virtual de Java instalada y el programa en su equipo
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El usuario abre el programa</li> <li>2. El usuario pulsa en el menú “Acerca de” dentro del apartado “Ayuda”</li> </ol>
<b>Verificación</b>	<ul style="list-style-type: none"> <li>• La información del programa se muestra correctamente</li> </ul>

Tabla 107: Caso de prueba CP-009

<b>ID CP-010</b>	
<b>Nombre</b>	Salir del programa
<b>Caso de uso</b>	CU-007
<b>Descripción</b>	El usuario ha realizado todas las operaciones que necesitaba y desea salir del programa
<b>Precondiciones</b>	El usuario debe tener el programa abierto
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa en el menú “Salir” dentro del apartado “Archivo” o</li> <li>1. El usuario pulsa en el botón de cerrar de la interfaz</li> </ol>
<b>Verificación</b>	<ul style="list-style-type: none"> <li>• La aplicación se cierra correctamente</li> </ul>

Tabla 108: Caso de prueba CP-010

La matriz de trazabilidad representada en la tabla 109 permite observar de una forma más visual la relación entre los casos de uso (representados en el eje horizontal) y los casos de prueba (representados en el eje vertical) del sistema.

ID	CU-001	CU-002	CU-003	CU-004	CU-005	CU-006	CU-007
CP-001	•		•	•			
CP-002		•	•	•			
CP-003		•	•	•			
CP-004	•		•	•			
CP-005	•		•	•			
CP-006	•		•	•			
CP-007	•				•		
CP-008		•			•		
CP-009						•	
CP-010							•

Tabla 109: Matriz de trazabilidad de los casos de uso y los casos de prueba

## Evaluación de usabilidad

Para la validación de la herramienta se optó por realizar un test de usabilidad a un grupo cerrado de usuarios. El concepto de usabilidad proviene del inglés “usability”. A pesar de no ser una palabra aceptada por la RAE, se usa a menudo en contextos informáticos y tecnológicos para hacer referencia a la facilidad de uso que tiene un entorno o una herramienta [USAB].

El test consistía en 12 preguntas de las cuales las dos primeras eran clasificatorias (preguntando por el género del encuestado y la experiencia en el campo del mismo, guardando el anonimato en todo momento) y el resto hacían referencia a la usabilidad real de la aplicación siendo todas ellas las siguientes:

1. Creo que usaré esta aplicación con frecuencia
2. Encontré la aplicación innecesariamente compleja
3. Pienso que la aplicación es fácil de usar
4. Creo que necesitaría el apoyo de un experto para usar la aplicación
5. Encontré las diversas posibilidades de la aplicación bien integradas
6. Pienso que hay demasiada inconsistencia en la aplicación
7. Imagino que la mayoría de las personas aprenderían muy rápidamente a utilizar la aplicación
8. Encontré la aplicación muy grande al recorrerla
9. Me sentí muy confiado en el manejo de la aplicación
10. Necesito aprender muchas cosas antes de manejarla con la aplicación

La selección de estas preguntas se hizo traduciendo la Escala de Usabilidad del Sistema (System Usability Scale o SUS en inglés) creada por John Brooke en el año 1996 [SUS] con la ayuda de un formulario creado en Google Forms, el cual puede encontrarse anexo a este documento [Anexo VI. Formulario de usabilidad. Este formulario debía responderse después de probar la aplicación y teniendo en cuenta como contexto el de un usuario objetivo del sistema.

## 5.2 Análisis de resultados

Después de presentar los métodos de evaluación usados, en este punto se presentan los resultados procedentes de la realización del proceso de verificación mediante el plan de pruebas y la validación por parte de los usuarios.

Como último paso del proceso de verificación del sistema, se deben comprobar uno por uno los casos de prueba presentados con el objetivo de comprobar el sistema. La tabla 110 presenta el resultado de cada uno de los casos de prueba del sistema después de comprobarlos.

Caso de prueba	Resultado	Fecha de realización
CP-001	Éxito	08/09/2015
CP-002	Éxito	08/09/2015
CP-003	Éxito	08/09/2015
CP-004	Éxito	08/09/2015
CP-005	Éxito	08/09/2015
CP-006	Éxito	08/09/2015
CP-007	Éxito	08/09/2015
CP-008	Éxito	08/09/2015
CP-009	Éxito	08/09/2015
CP-010	Éxito	08/09/2015

Tabla 110: Resultados de la verificación del sistema

Como se puede ver, todos los casos de prueba se han conseguido superar con éxito por lo que el sistema queda verificado. Para mayor seguridad sobre su validez y encontrar posibles puntos flacos de cara al uso por parte de personas reales, el formulario de usabilidad realizado ayuda a esclarecerlo.

Un total de once personas respondieron el cuestionario. De estas once, teniendo en cuenta los datos observados, una de ellas respondió de forma sospechosa a todas las preguntas con el máximo valor generando incongruencias en sus respuestas, por lo que se eliminaron del estudio dejando un total de diez usuarios encuestados.

Para la valoración de las preguntas se usó una escala del uno al cinco siendo el uno “En completo desacuerdo” y el cinco “Completamente de acuerdo”. Cabe destacar tal y como se ve en la figura 34 que la mayoría de los usuarios apenas contaba con experiencia en el campo, lo que implica que los resultados del estudio estarán adaptados a audiencias de este tipo y no tanto a usuarios avanzados. En cuanto al sexo los resultados fueron bastante equitativos correspondiendo los datos a cuatro mujeres y seis hombres encuestados.

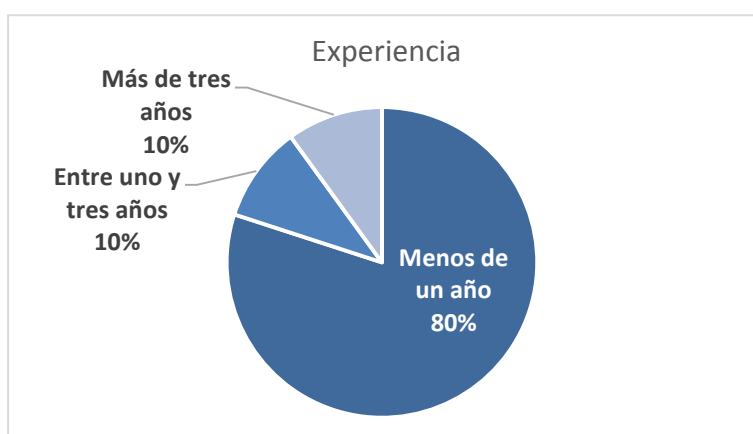


Figura 34: Experiencia de los usuarios encuestados

Entrando en el análisis de los datos referentes a las preguntas, se expondrá una gráfica por cada una de ellas seguido de un breve análisis. En la figura 35 se aprecia la primera pregunta.

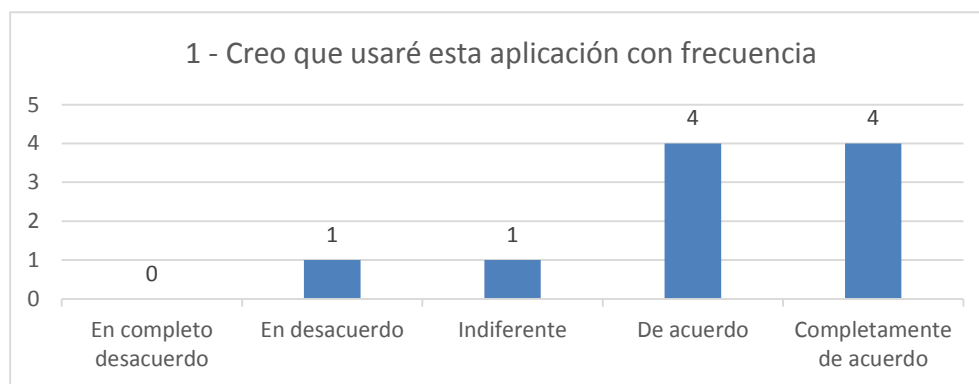


Figura 35: Resultados de la primera pregunta

Del total de usuarios, un 80% considera que usará la aplicación en el futuro, siendo la mitad de estos, un 40% del total usuarios que lo consideran por completo. De los otros dos restantes sólo uno de ellos está en desacuerdo.

Las siguientes dos preguntas tienen cierta relación ya que preguntan de formas distintas sobre la complejidad que ha percibido el usuario al probar la aplicación. Se pueden observar las respuestas en las figuras 36 y 37.

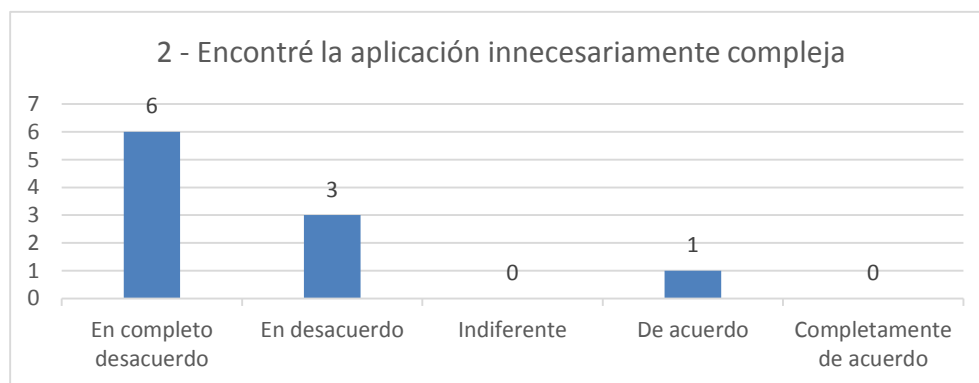


Figura 36: Resultados de la segunda pregunta

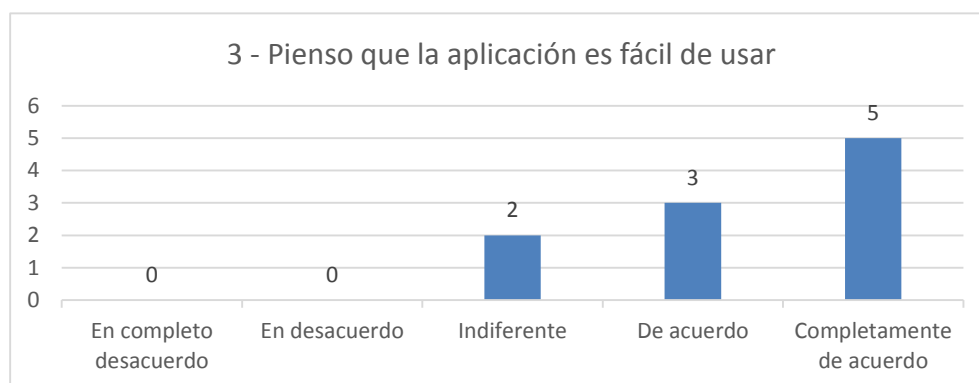


Figura 37: Resultados de la tercera pregunta

En la pregunta dos, el 90% de los usuarios estaba en desacuerdo con que la aplicación fuera innecesariamente compleja, estando un 60% del total en completo desacuerdo, por lo que se puede considerar que la aplicación es compleja en su justa medida. En el caso de la tercera pregunta, un 80% de los usuarios consideraron que la aplicación era fácil de usar, considerando un 50% del total que era muy sencilla.

La cuarta pregunta reflejada en la figura 38 también se refiere en cierto modo a la dificultad preguntando sobre la necesidad de un experto para asesorar y enseñar sobre su uso.

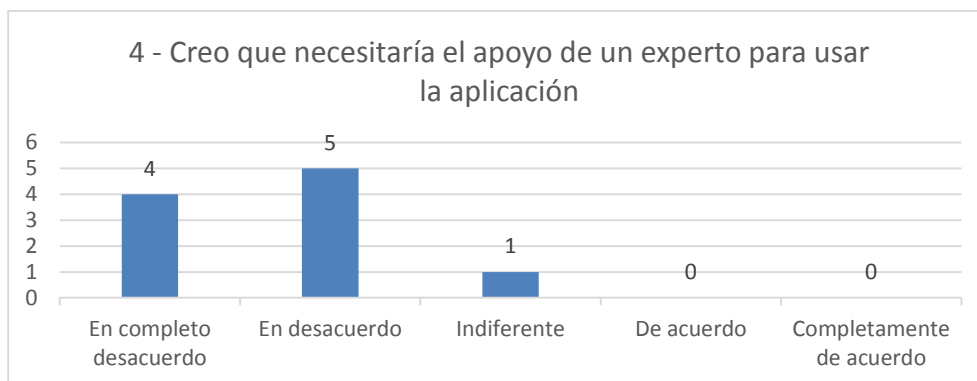


Figura 38: Resultados de la cuarta pregunta

El 90% de los encuestados consideraron que podían hacer uso de la herramienta sin ayuda de un experto, estando el grueso de las respuestas (un 50%) en simple desacuerdo con la necesidad del mismo.

Las siguientes dos preguntas representadas en las figuras 39 y 40 hacen referencia a la efectividad de la integración de las características del programa.

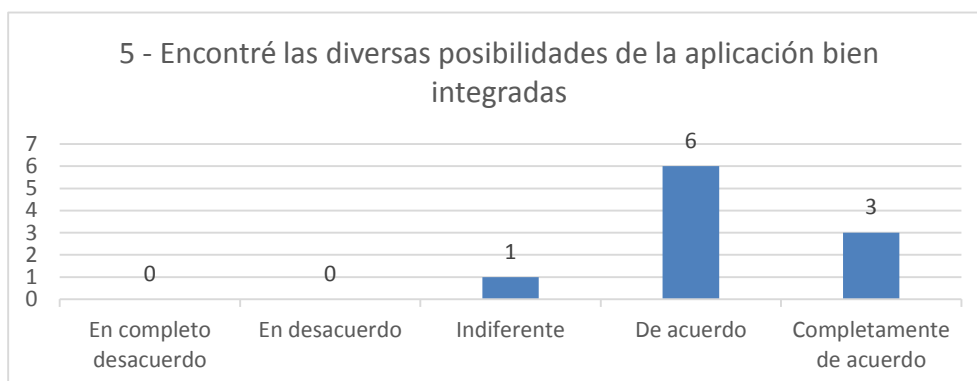


Figura 39: Resultados de la quinta pregunta

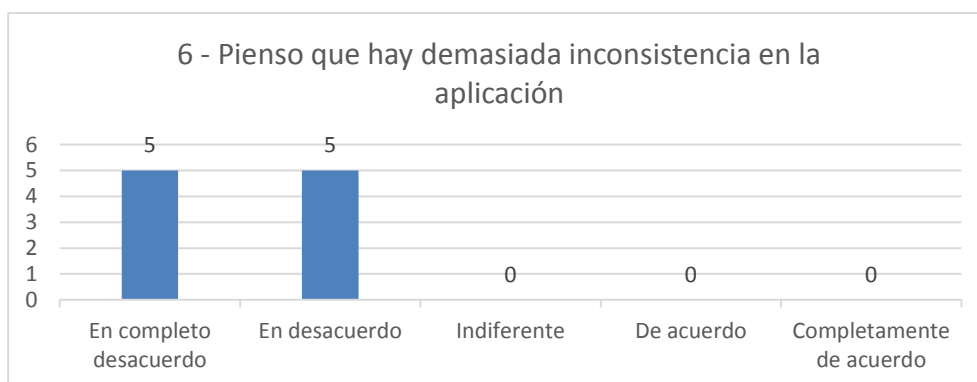


Figura 40: Resultados de la sexta pregunta



La quinta pregunta encuestaba sobre el nivel de integración de las características en la versión final, pregunta a la que un 90% de los encuestados respondió que era óptimo aunque sólo un 30% del total le pareció que no había margen de mejora. En la sexta pregunta el 100% de los encuestados consideraron que la aplicación no presentaba inconsistencia, estando repartidas las respuestas por igual entre “En completo desacuerdo” y “En desacuerdo”.

En la figura 41 se representan las respuestas de los usuarios a la séptima pregunta.

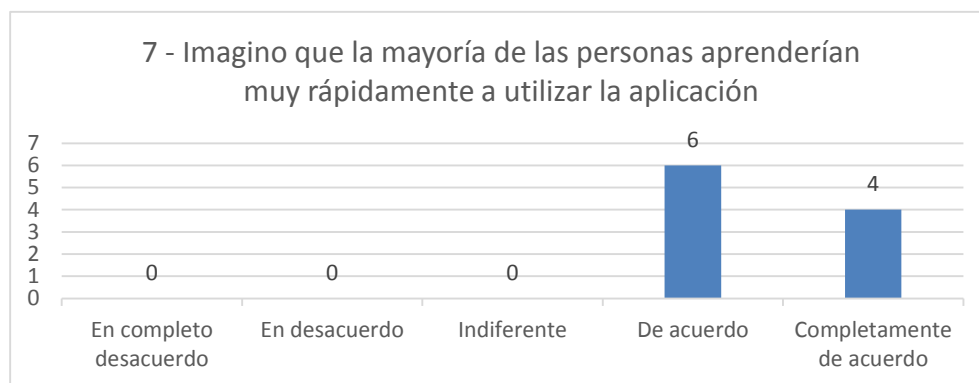


Figura 41: Resultados de la séptima pregunta

En esta ocasión, el 100% de los usuarios opinan que otras personas serían capaces de aprender a utilizarla sin problemas, siendo un 40% de los usuarios los que están de acuerdo por completo.

La figura 42 representa las respuestas de la octava pregunta.

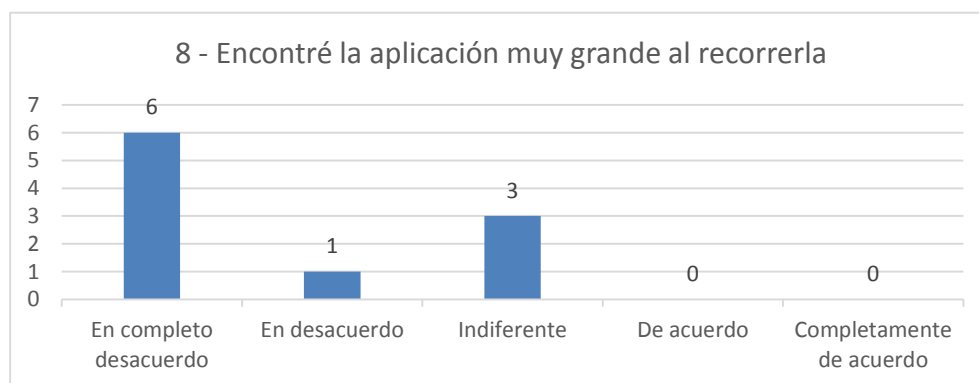


Figura 42: Resultados de la octava pregunta

El 70% de los usuarios consideraron que la aplicación no era grande en exceso al recorrerla habiendo respondido un 60% del total que estaban en completo desacuerdo con ello. El 30% restante no se posicionó, indicando que estaba en su justa medida.

Las últimas dos preguntas hacen referencia a la facilidad de manejo de la aplicación y se reflejan en las figuras 43 y 44.

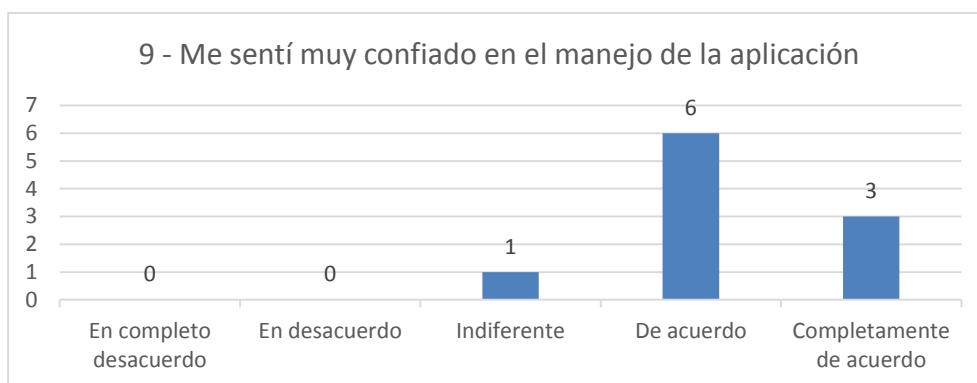


Figura 43: Resultados de la novena pregunta

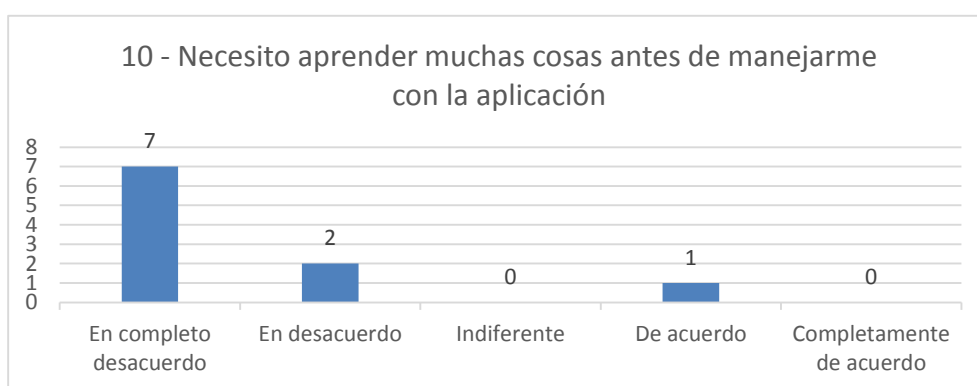


Figura 44: Resultados de la décima pregunta

En el caso de la pregunta nueve, los encuestados aseguraron con un 90% de apoyo que se sentían confiados en el manejo de la aplicación, siendo un 30% del total los que se decantaron por la opción “Completamente de acuerdo” y un 10% los que se mostraron indiferentes. En la décima y última pregunta al igual que en la anterior un 90% de los usuarios se mostraron en desacuerdo con la necesidad de tener un periodo de aprendizaje extenso previo al uso de la aplicación, correspondiendo un 70% a la opción “En completo desacuerdo”. El 10% restante estaba de acuerdo con esa afirmación.

Observando todos estos resultados, se puede decir que por parte de los usuarios la aceptación de la aplicación final es bastante elevada, consiguiendo siempre que el grueso de las respuestas recaigan sobre las opciones positivas o muy positivas. Existen también puntos de mejora que se deben tener en cuenta para las siguientes versiones.

El proceso completo de evaluación mediante las técnicas de verificación y validación, concluye que la herramienta está lista para ser usada pues cumple con los requisitos marcados en las primeras fases del desarrollo.

## 6 Conclusión

---

En este último capítulo se resume el propósito del trabajo y las principales aportaciones realizadas, así como los posibles trabajos futuros y baches encontrados durante el desarrollo del trabajo. Para concluir se resumirán una serie de apreciaciones personales.

### 6.1 Propósito del trabajo y aportaciones realizadas

---

El desarrollo del proyecto ha aportado la construcción de una aplicación para ayudar a extraer y tratar datos de consumo eléctrico. La principal aportación de esta aplicación reside en la posibilidad de extraer un gran volumen de datos de agregación diaria de forma eficiente, recurso que no existía ni en la web ni en posibles programas de la competencia hasta la construcción de la herramienta.

Adicionalmente a esta aportación, el proyecto también permite exportar todos los datos recabados de forma pretratada para su posterior aprovechamiento en programas de cálculo del tipo MatLab entre otro tipo de extensiones admitidas, así como visualizar de forma sencilla cada uno de los informes generados.

En otro orden de cosas, otra de las aportaciones concretas de la herramienta reside en su finalidad, es decir, la herramienta que se ha generado durante la consecución del proyecto, es un programa final y acabado que satisface las necesidades de un cliente real y será usado por el departamento de Ingeniería Eléctrica de la Universidad Carlos III como ayuda para la extracción de los datos necesarios para sus investigaciones e incluso para la docencia de este departamento.

### 6.2 Trabajos futuros

---

Como en todo desarrollo, hay características opcionales o deseables que se quedan fuera del producto final por falta de recursos, ya sean humanos, tecnológicos, económicos o temporales. Durante el desarrollo de la herramienta se plantearon varias características que sería deseable incluir si bien no estaban pactadas en la toma de requisitos ni son indispensables per se.

La primera de ellas es la traducción del programa a distintos idiomas o como mínimo a inglés. Esta opción se dejó fuera por falta del manejo del sistema de referencias a cadenas y su sustitución en JavaFX.

En las primeras iteraciones del programa también se manejó la idea de dar la posibilidad al usuario final de abrir archivos de datos anteriormente exportados desde la misma herramienta. La idea se desechó pronto debido al tiempo limitado del que se disponía por no suponer una característica indispensable al ser posible siempre recuperar el informe volviendo a generarlo. Sería deseable su inclusión para poder analizar los datos exportados en caso de no contar con una conexión a internet a través de la cual realizar las peticiones a la web de Red Eléctrica en un momento dado. En cuanto al guardado de archivos, sería deseable dar más libertad al usuario sobre que quiere guardar exactamente, es decir, en caso de realizar un informe por rangos de fechas, dejar decidir al usuario si lo quiere exportar al completo, si quiere acotarlo o si quiere uno sólo de todos los solicitados.

En las fases finales de desarrollo, se intentó implementar una interfaz distinta más elaborada. Debido precisamente a que era más elaborada se desechó, pero se considera una mejora para futuras iteraciones realizar una reorganización completa de la interfaz del programa.

A pesar de que la aplicación actúa como sustituta de la web de la que extrae datos, no todos los informes presentes en esta se encuentran incluidos en la herramienta, sólo los exigidos por el cliente (los más usados o más importantes) se encuentran implementados. Sería una buena inclusión para

el futuro la adición de todos los informes existentes para poder prescindir por completo de la web de Red Eléctrica.

Aunque desde un primer momento se planteó la aplicación como una aplicación nativa, dada la naturaleza de Java es posible portarla a aplicación web o aplicación móvil (sobre todo para Android dónde el lenguaje de programación es Java). Esto es más una idea que algo deseable ya que el propio carácter de la herramienta la ata a ordenadores personales donde el manejo de los datos extraídos sea posible, pero a nivel de visualización o control de los mismos podría ser útil. Al igual que otras características aquí comentadas carecen de prioridad y sería decisión del cliente su necesidad futura.

Puesto que el usuario final siempre tiene la última palabra, sería aconsejable realizar encuestas de satisfacción a los clientes con tal de saber cuáles son los puntos flacos a la hora del uso de la herramienta. De esta forma se podría saber con seguridad el camino a seguir para posibles mejoras o implementaciones adicionales al desarrollo final pactado y evaluado.

### 6.3 Problemas encontrados

---

Durante el desarrollo de la herramienta se han encontrado problemas al igual que ocurre en todo tipo de proyectos. En la fase de implementación de la herramienta se ha tenido que lidiar con distintos tipos de tecnologías y herramientas, algunas viejas conocidas como Java o Eclipse y otras totalmente nuevas como JavaFX.

El uso de tecnologías por conocer siempre supone un bache en el camino al no poder avanzar tan rápido como en un principio se desearía. Sobre JavaFX cabe destacar que si bien está creado por Oracle para construir interfaces elaboradas en Java con poco esfuerzo, los primeros pasos son sencillos pero a medida que se aumenta la complejidad de la interfaz, la dificultad de la implementación de la misma aumenta en mayor escala.

La escasa documentación existente sobre esta herramienta para generar interfaces y sobre todo la poca información que dan los logs cuando se produce una excepción genera frustraciones y retrasos que serían fácilmente evitables si la información aportada por la herramienta fuera más esclarecedora o fuera más sencillo encontrar documentación sobre errores.

El hecho de que exista una herramienta para crear archivos de interfaz con un simple arrastrar y soltar es una forma cómoda de empezar y esbozar el resultado que se espera y se tiene ideado, pero a la hora de terminar y conectar la interfaz no hay herramientas de ayuda y una vez más existe el problema de la falta de información sobre los errores por parte de JavaFX y la falta de documentación. Las búsquedas en internet alivian estas carencias pero sería oportuno que se diese más soporte por parte de Oracle.

Por último cabe mencionar que el más lastrante de todos los problemas y presente en todo momento, ha sido la falta de tiempo debido a la compatibilización del proyecto universitario con la jornada laboral. Los problemas personales también han constituido un obstáculo a la hora de la realización del trabajo. Si bien estos problemas son comunes y afectan a cualquiera que se proponga llevar a cabo un proyecto, merecen mención para explicar la dilatación de la finalización del mismo.

### 6.4 Opiniones personales

---

La realización de este proyecto supone la culminación de años de trabajo y estudio. Todos nos enfrentamos a hitos importantes en nuestras vidas, siendo la finalización de una carrera universitaria uno de los grandes sobre todo en cuanto a realización personal. La culminación de este proyecto es uno de los momentos más remarcables sólo comparable en cuanto a nivel académico

al momento en el que abres el correo que contiene la nota de la última asignatura que te falta por aprobar.

A pesar de contar con trabajo estable desde hace tres años, este proyecto me ha ayudado a entender el mundo del software más en profundidad, ya que en el entorno real, son pocas las veces que se hace una documentación tan extensa del trabajo realizado o un seguimiento tan exhaustivo del propio proyecto y su planificación en sí. Otra de las diferencias sustanciales es que mi entorno laboral se centra en el desarrollo web al completo, trabajando en las áreas de backend y frontend, donde la realización de las interfaces se basa en diseño HTML o de otros lenguajes propios de la web. De esta forma este proyecto me ha enseñado a realizar una interfaz nativa desde cero usando una librería que desconocía hasta el momento pero que es lo suficientemente potente como para poder considerarla en mis proyectos futuros, tanto personales como laborales.

Algunos puntos tales como la conectividad con los servidores de Red Eléctrica son parte de mi día a día por lo que no me han supuesto un reto, al igual que el trato con el cliente real, pues es algo a lo que estoy acostumbrado.

Como crítica personal, se me ocurre que el tiempo de realización del proyecto al completo podría haber sido menor. Este trabajo me ha enseñado a organizar el poco tiempo disponible de una forma más óptima, lo cual me ayudará en el futuro. Sin nada más que añadir, estoy contento con el trabajo realizado pues he conseguido llevar adelante un proyecto con un cliente y una utilidad real hasta su finalización, habiendo quedado este verificado y validado.

## 7 Bibliografía

En este apartado se muestran todas las referencias que han sido usadas para la creación de la presente memoria ordenadas de forma alfabética con respecto a su identificador.

- [ADOCR] Página de producto de Adobe Acrobat. Disponible [Internet]: <<https://acrobat.adobe.com/es/es/how-to/ocr-software-convert-pdf-to-text.html>> [25 de septiembre de 2015]
- [AGIL] Navarro Cadavid, A., Fernández Martínez, J.D. y Morales Vélez, J.: 'Revisión de metodologías ágiles para el desarrollo de software'. Universidad Icesi, 2013, <<http://dialnet.unirioja.es/descarga/articulo/4752083.pdf>> [25 de septiembre de 2015]
- [ANT] Página de producto de Apache Ant. Disponible [Internet]: <<http://ant.apache.org/>> [25 de septiembre de 2015]
- [AWT] Documentación de AWT. Disponible [Internet]: <<https://docs.oracle.com/javase/7/docs/technotes/guides/awt/>> [25 de septiembre de 2015]
- [C#] Página de producto de C#. Disponible [Internet]: <<https://msdn.microsoft.com/es-es/library/z1zx9t92.aspx>> [25 de septiembre de 2015]
- [C++] Página del estándar C++. Disponible [Internet]: <<https://isocpp.org/>> [25 de septiembre de 2015]
- [CONSUMO] Hábitos de consumo eléctrico. Disponible [Internet]: <<http://www.ree.es/es/red21/eficiencia-energetica-y-consumo-inteligente/nuestros-habitos-de-consumo>> [25 de septiembre de 2015]
- [CONTROLSFX] Página de producto de ControlsFX. Disponible [Internet]: <<http://fxexperience.com/controlsfx/>> [25 de septiembre de 2015]
- [DTASCR] Definición de data scraper. Disponible [Internet]: <[https://en.wikipedia.org/wiki/Data\\_scraping](https://en.wikipedia.org/wiki/Data_scraping)> [25 de septiembre de 2015]
- [ECLIPSE] Página de producto de Eclipse. Disponible [Internet]: <<https://eclipse.org/home/index.php>> [25 de septiembre de 2015]
- [ENDDBB] Ling Liu, M. Tamer Özsu: 'Encyclopedia of Database Systems' (Springer, 2009, 1ª edición), pág. 2511-2512. Disponible [Internet]: <<http://rd.springer.com/book/10.1007/978-0-387-39940-9>> [25 de septiembre de 2015]
- [ESIOS] Sistema de información del operador del sistema (Red Eléctrica Española). Disponible [Internet]: <<http://www.esios.ree.es/web-publica/>> [25 de septiembre de 2015]
- [GESTION] Técnicas de gestión de la demanda. Disponible [Internet]: <<http://www.ree.es/es/actividades/operacion-del-sistema/gestion-de-demanda>> [25 de September de 2015]
- [HAP] Página de producto de HTML Agiliti Pack. Disponible [Internet]: <<http://htmlagilitypack.codeplex.com/>> [25 de September de 2015]
- [IMAC] Página de producto de iMacros. Disponible [Internet]: <<http://imacros.net/>> [25 de September de 2015]
- [IMIO] Página de producto de import.io. Disponible [Internet]: <<https://import.io/>> [25 de September de 2015]
- [INGELEC] Página del departamento de Ingeniería Eléctrica de la Universidad Carlos III de Madrid. Disponible [Internet]: <[http://portal.uc3m.es/portal/page/portal/dpto\\_ing\\_electrica/](http://portal.uc3m.es/portal/page/portal/dpto_ing_electrica/)> [25 de September de 2015]

- [ISOCICLO] Página de definición del estándar sobre el ciclo de vida de los procesos de software. Disponible [Internet]: <[http://www.iso.org/iso/catalogue\\_detail?csnumber=43447](http://www.iso.org/iso/catalogue_detail?csnumber=43447)> [25 de September de 2015]
- [JAVA] Página de producto de Java. Disponible [Internet]: <<https://www.oracle.com/java/technologies/java-se.html>> [25 de September de 2015]
- [JFC] Página de definición de las Java Foundation Classes. Disponible [Internet]: <<http://www.oracle.com/technetwork/java/faqs-140150.html>> [25 de September de 2015]
- [JFX] Página de producto de JavaFX. Disponible [Internet]: <<http://docs.oracle.com/javafx/2/overview/jfxpub-overview.htm>> [25 de September de 2015]
- [JSOUP] Página de producto de jsoup. Disponible [Internet]: <<http://jsoup.org/>> [25 de September de 2015]
- [KNWGRP] Página de presentación de Google Knowledge Graph. Disponible [Internet]: <<http://googleblog.blogspot.com.es/2012/05/introducing-knowledge-graph-things-not.html>> [25 de September de 2015]
- [LOGAMZ] Página del ratón Logitech B100 en Amazon. Disponible [Internet]: <<http://www.amazon.es/Logitech-B100-Rat%C3%B3n-%C3%B3ptico-negro/dp/B00AZKNPZC/>> [25 de September de 2015]
- [MATLAB] Página de producto de la aplicación MatLab. Disponible [Internet]: <<http://es.mathworks.com/products/matlab/>> [25 de September de 2015]
- [MVC] Transparencias sobre Modelo Vista Controlador. Disponible [Internet]: <<https://www.fdi.ucm.es/profesor/jpavon/poo/2.14.MVC.pdf>> [25 de September de 2015]
- [OXDICT] Definición de “scrape” en el Oxford Dictionary. Disponible [Internet]: <<http://www.oxforddictionaries.com/es/traducir/ingles-espanol/scrape>> [25 de septiembre de 2015]
- [PYTH] Página de producto de Python. Disponible [Internet]: <<http://www.python.org/>> [25 de September de 2015]
- [SAL] Página para calcular salarios aproximados por puesto y experiencia. Disponible [Internet]: <<http://www.tusalario.es/main/salario/comparatusalario>> [25 de septiembre de 2015]
- [SCRSTE] Definición de scraper site. Disponible [Internet]: <[https://en.wikipedia.org/wiki/Scraper\\_site](https://en.wikipedia.org/wiki/Scraper_site)> [25 de septiembre de 2015]
- [SCRWI] Página de producto de Scraper Wiki. Disponible [Internet]: <<https://scraperwiki.com/>> [25 de September de 2015]
- [SEGCAT] Categorías de cotización de la seguridad social. Disponible [Internet]: <[http://www.seg-social.es/Internet\\_1/Trabajadores/CotizacionRecaudaci10777/Basesytiposdecotiza36537/index.htm](http://www.seg-social.es/Internet_1/Trabajadores/CotizacionRecaudaci10777/Basesytiposdecotiza36537/index.htm)> [25 de September de 2015]
- [SEGCOT] Bases de cotización de la seguridad social. Disponible [Internet]: <[http://www.seg-social.es/Internet\\_1/Trabajadores/CotizacionRecaudaci10777/Regimenes/RegimenGeneraldeS10957/InformacionGeneral/index.htm](http://www.seg-social.es/Internet_1/Trabajadores/CotizacionRecaudaci10777/Regimenes/RegimenGeneraldeS10957/InformacionGeneral/index.htm)> [25 de September de 2015]
- [SIKULI] Sikuli. Sikuli Lab, University of Colorado Boulder. 2009. Disponible [Internet]: <<http://www.sikuli.org/>> [25 de septiembre de 2015]
- [SUS] Brooke, J.: ‘SUS-A quick and dirty usability scale. Usability evaluation in industry’, 1996, vol. 189, nº 194, págs. 4-7. Disponible [Internet]:

- <<https://books.google.es/books?id=IfUsRmzAqvEC&pg=PA189>> [25 de September de 2015]
- [SWING] Documentación de Swing. Disponible [Internet]: <<https://docs.oracle.com/javase/7/docs/technotes/guides/swing/index.html>> [25 de September de 2015]
- [TEAM] Página de producto de TeamViewer. Disponible [Internet]: <<https://www.teamviewer.com/es/index.aspx>> [25 de September de 2015]
- [TIOBE] Lenguajes de programación más usadas. Disponible [Internet]: <<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>> [25 de September de 2015]
- [TOSAMZ] Página del portátil Toshiba Satellite en Amazon. Disponible [Internet]: <<http://www.amazon.es/Toshiba-Satellite-L50D-B-14U-Port%C3%A1til-Quad-Core/dp/B00MGR6VXU/>> [25 de September de 2015]
- [USAB] Definición de usabilidad. Disponible [Internet]: <<http://definicion.de/usabilidad/>> [25 de September de 2015]
- [VOLERE] Plantilla de requisitos estandarizada. Disponible [Internet]: <[http://www.volere.co.uk/pdf%20files/template\\_es.pdf](http://www.volere.co.uk/pdf%20files/template_es.pdf)> [25 de September de 2015]
- [WEX] Página de producto de Web Extractor. Disponible [Internet]: <<http://www.webextractor.com/>> [25 de septiembre de 2015]
- [WSIO] Página de producto de webscraper.io. Disponible [Internet]: <<http://webscraper.io/>> [25 de septiembre de 2015]



## Anexo I. Control de versiones

En este anexo se muestran las versiones que ha atravesado el presente documento:

<b>Versión</b>	<b>Fecha</b>	<b>Páginas</b>	<b>Procesador</b>	<b>Cambios</b>
<b>0.1.0</b>	15/06/2015	23	Microsoft Word 2013	Primera versión del documento, creación de la estructura de la memoria
<b>0.2.0</b>	8/07/2015	46	Microsoft Word 2013	Incorporación del apartado “Estado del arte”
<b>0.2.0 (Rev)</b>	13/07/2015	46	Microsoft Word 2013	Revisión de la versión 0.2.0
<b>0.2.1</b>	16/07/2015	44	Microsoft Word 2013	Corrección de los errores encontrados en la versión 0.2.0
<b>0.3.0</b>	22/07/2015	46	Microsoft Word 2013	Incorporación del apartado “Gestión de proyecto software”
<b>0.3.0 (Rev)</b>	28/07/2015	46	Microsoft Word 2013	Revisión de la versión 0.3.0
<b>0.3.1</b>	28/07/2015	46	Microsoft Word 2013	Corrección de los errores encontrados en la versión 0.3.0
<b>0.4.0</b>	31/08/2015	96	Microsoft Word 2013	Incorporación del apartado “Solución”
<b>0.4.0 (Rev)</b>	1/09/2015	96	Microsoft Word 2013	Revisión de la versión 0.4.0
<b>0.4.1</b>	6/09/2015	102	Microsoft Word 2013	Corrección de los errores encontrados en la versión 0.4.0
<b>0.5.0</b>	13/09/2015	121	Microsoft Word 2013	Incorporación del apartado “Evaluación” y el anexo V
<b>0.5.0 (Rev)</b>	14/09/2015	119	Microsoft Word 2013	Revisión de la versión 0.5.0
<b>0.5.1</b>	14/09/2015	120	Microsoft Word 2013	Corrección de los errores encontrados en la versión 0.5.0
<b>0.6.0</b>	16/09/2015	127	Microsoft Word 2013	Incorporación de los apartados “Introducción”, “Conclusión”, tablas de definiciones y el anexo VI
<b>0.6.0 (Rev)</b>	17/09/2015	129	Microsoft Word 2013	Revisión de la versión 0.6.0
<b>0.6.1</b>	17/09/2015	126	Microsoft Word 2013	Corrección de los errores encontrados en la versión 0.6.0
<b>0.7.0</b>	19/09/2015	129	Microsoft Word 2013	Incorporación del resumen, el abstract, los agradecimientos y los anexos II, III y IV Arreglados los estilos y los índices
<b>0.7.0 (Rev)</b>	21/09/2015	129	Microsoft Word 2013	Revisión de la versión 0.7.0
<b>1.0.0</b>	24/09/2015	129	Microsoft Word 2013	Versión final del documento

Tabla 111: Versiones del documento

Anexo II. Diagrama del desarrollo de la herramienta

En este anexo se muestra a modo de ejemplo un diagrama de Gantt del desarrollo de la última iteración de la herramienta (no de su planificación), desde la última entrevista hasta el resultado final, dividiéndolo en tareas pequeñas. El eje horizontal se divide en días mientras que el vertical se divide en tareas.

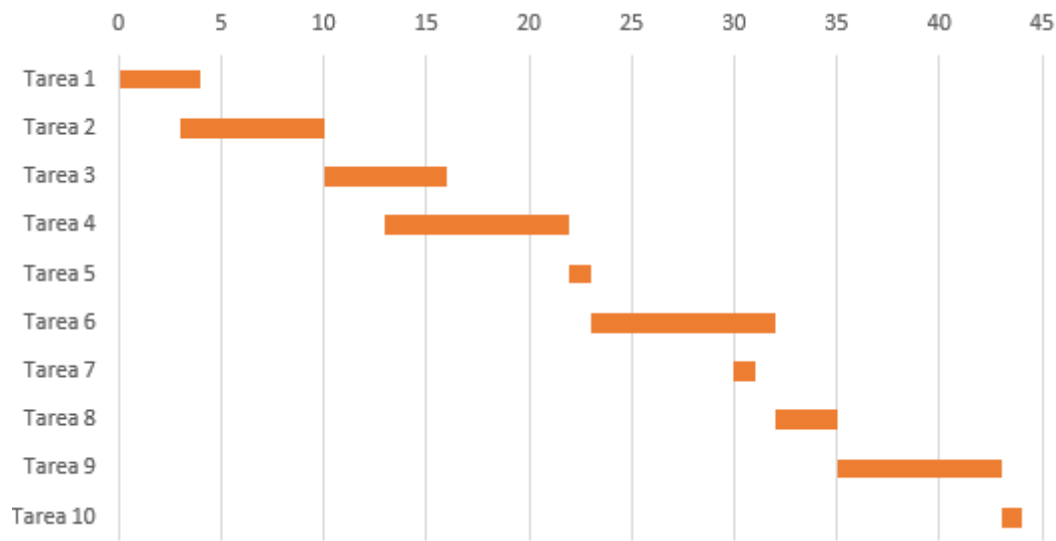


Figura 45: Diagrama de Gantt de la última iteración

Tarea	Descripción
1	Creación de clase genérica para respuestas de todos los informes en lugar de tener una clase por cada tipo de informe
2	Cambio de campos de texto por tablas
3	Guardado de archivos
4	Vista genérica para todos los informes en lugar de tener una vista por informe
5	Nuevos informes añadidos
6	Incorporada la vista de gráfica
7	Cambios en el menú de la aplicación
8	Reordenación de la interfaz de la perspectiva de respuesta
9	Arreglo de bugs finales
10	Creación del ejecutable

Tabla 112: Tareas de la última iteración

### Anexo III. Entrevistas con el cliente

---

Puesto que este proyecto cuenta con un cliente real, el departamento de Ingeniería Eléctrica de la UC3M, se realizaron varias entrevistas desde el inicio del mismo al igual que se haría en un proyecto comercial o en la vida real.

La primera de las entrevistas, la toma de contacto con el cliente siguió el siguiente esquema:

#### Guion de la entrevista del 24-1-2014

Aunque es algo ya sabido, quizá sería buena idea empezar la entrevista hablando de forma general sobre el objetivo y el alcance del proyecto para saber cuáles son las expectativas del cliente y en qué grado se van a poder satisfacer con los recursos disponibles (tiempo, conocimientos...).

#### *ALCANCE DEL PROYECTO*

- ¿Cuáles son las necesidades del cliente?
- ¿En qué medida podemos cubrirlas con los recursos disponibles?
- ¿Qué expectativas tiene el cliente?

#### *EXTRACCIÓN DE REQUISITOS*

Sería interesante establecer desde un principio las limitaciones y las características mediante una obtención de requisitos, ya sea en la primera o en las entrevistas sucesivas.

- ¿Cuál es el objetivo de la obtención de datos?
- ¿Qué tipo de datos debemos recoger?
- ¿Cuáles son los principales atributos/dimensiones de interés de los mismos?
- ¿En qué intervalos de tiempo deben recogerse los datos?
- ¿Cómo deben mostrarse los datos al usuario?
- ¿Es necesario realizar algún tratamiento sobre los datos?
- En caso de que sea así, ¿se persigue un tratamiento automático o interactivo, realizado por el usuario final (por ejemplo, basado en aplicación de técnicas de filtrado, selección, etc.)?
- ¿Deben guardarse los datos recogidos?
- En caso de que deban guardarse, ¿qué formatos deberían considerarse?
- ¿Existe alguna restricción referente a la plataforma de despliegue (por ejemplo, sistema operativo)?

#### *PLANIFICACIÓN*

Para terminar, habiendo establecido ya las bases del proyecto, sería conveniente pactar los intervalos temporales para cada fase del desarrollo. En este aspecto las limitaciones vienen dadas por la entrega y presentación del proyecto, pero podría hacerse algún planning dentro de este intervalo de tiempo.

Tras pactar estos aspectos se realizaron reuniones de seguimiento posteriores en las que no se realizó un guion concreto, siendo simplemente para guiar el desarrollo. En ellas se realizaron cambios sobre los requisitos tomados inicialmente tales como la inclusión de las peticiones por intervalos. Tras varios prototipos se llegó a la versión final.

## Anexo IV. Manual de usuario

---

REWS (Red Eléctrica Web Scraper) es una aplicación diseñada con el fin de proporcionar al usuario una vía eficaz para la obtención de datos de consumo de la Red Eléctrica Española y su posterior procesamiento.

La aplicación se caracteriza por su sencillez y apenas es necesario tiempo para aprender a utilizarla. Durante el presente manual se indicarán las opciones que están disponibles y las acciones necesarias para llevarlas a cabo. Todas las ilustraciones, corresponden a la ejecución de la aplicación en Windows 8.1.

### Requisitos mínimos

Para la ejecución de la aplicación se deben cumplir ciertos requisitos mínimos de hardware y de software que se presentan a continuación.

#### *Hardware*

La única limitación específicamente de hardware viene definida por la resolución de pantalla cuyos valores mínimos para la correcta visualización es de 800x600 píxeles, resolución que en la actualidad se considera obsoleta por lo que no debe ser un impedimento para su correcta ejecución. A parte de esto, es necesario contar con una conexión a internet sobre la que poder realizar las peticiones para generar los informes.

#### *Software*

REWS es una aplicación basada íntegramente en Java, lo que implica que es una aplicación portable y puede ser usada en cualquier sistema operativo para el que exista Java. Para su correcta utilización es necesario contar con la máquina virtual de Java instalada en el equipo en su versión 8 update 20 o superior. Adicionalmente la aplicación se sirve de dos librerías externas proporcionadas junto al ejecutable cuya estructura de carpetas debe permanecer tal y como se proporciona para su correcto funcionamiento.

### Bienvenido a la aplicación

Una vez cumplidos los requisitos y ejecutada la aplicación se recibe al usuario con la sección de bienvenida. En esta sección se encuentra un sencillo diagrama del funcionamiento de la aplicación y unas indicaciones para comenzar con la creación de informes.

La aplicación cuenta con una interfaz lo más estándar posible que se adapta al sistema operativo en el que se ejecuta con el fin de que el usuario se sienta familiarizado desde el primer momento. Para navegar entre las distintas opciones, existe un menú en la parte superior que da acceso a todas las secciones de las que dispone el programa. En la sección de bienvenida representada en la figura 46 se pueden apreciar todos los detalles expuestos.

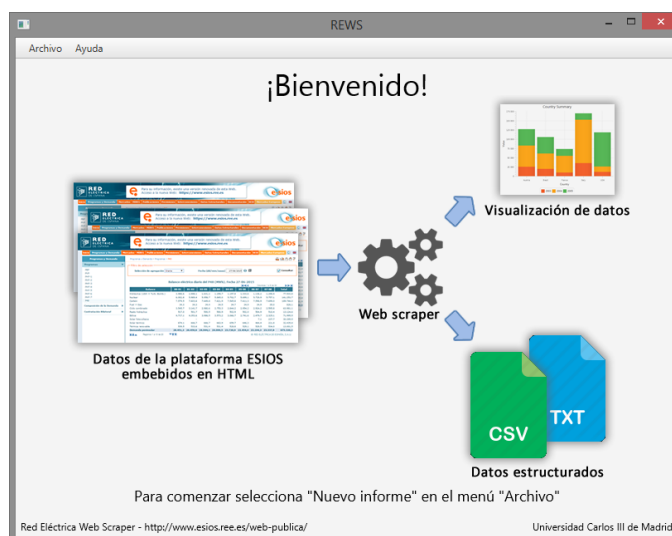


Figura 46: Sección de bienvenida

## Generando informes

Para acceder a la sección de generación de informes, el usuario debe hacer clic en el menú “Archivo” y más tarde, en el menú emergente en la opción “Nuevo informe”. Al realizar este proceso se mostrará una nueva sección representada en la figura 47.

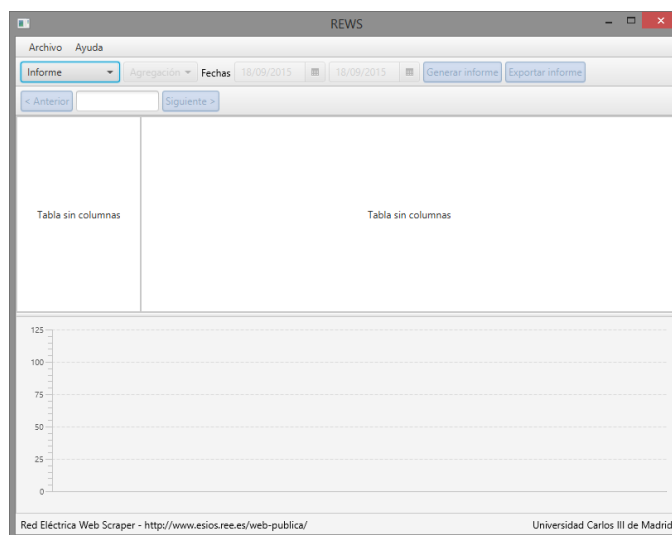


Figura 47: Sección de generación de informes

Por el momento, la sección permanece vacía esperando a que el usuario seleccione todas las opciones que se le ofrecen y genere un informe. Para ello, se debe seleccionar el tipo de informe deseado, la agregación deseada y las fechas finales e iniciales. En la figura 48 se representa la barra de menú con estas opciones ya seleccionadas para el informe de la demanda P48 con agregación diaria para un solo día.

Demanda P48 ▾
 Diaria ▾
 Fechas 28/03/2015 28/03/2015

Figura 48: Parámetros del informe

Una vez seleccionados estos datos, se debe pulsar el botón “Generar informe” situado a la derecha de los mismos. La apariencia de la herramienta después de pulsar el botón será la representada en la figura 49.

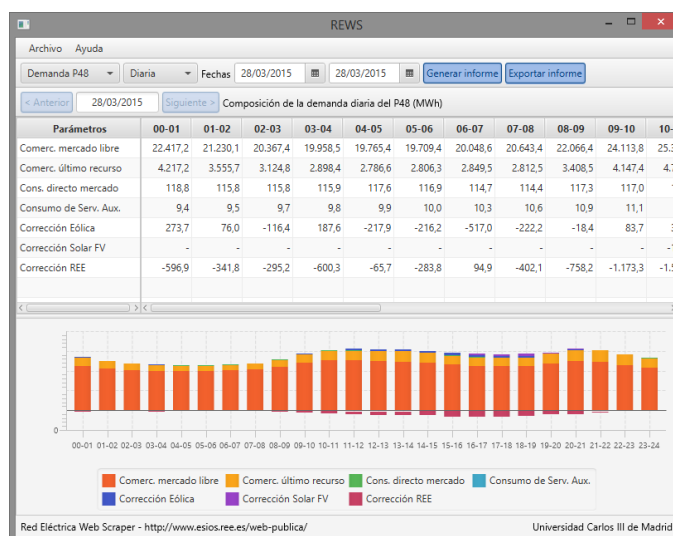


Figura 49: Apariencia de la aplicación tras realizar el informe

### Tipos de informes

En la aplicación existen diez tipos de informes que se pueden generar:

- Composición de la demanda del P48
- Balance eléctrico del P48
- Composición de la demanda del PBF
- Balance eléctrico del PBF
- Composición de la demanda del PVP
- Balance eléctrico del PVP
- Energía gestionada en el PBF mediante contratos bilaterales
- Previsión de la producción eólica nacional peninsular
- Energía Gestionada en el Mercado Diario en España
- Precios Marginales del Mercado Diario en España y Portugal

Todos ellos se pueden seleccionar desde el menú superior de la sección de generación e informes y en todos existen todas las agregaciones salvo en la previsión eólica y los precios marginales que disponen únicamente de agregación diaria.

### Tipos de agregación

La aplicación permite consultar todos los informes en todas las agregaciones salvo las dos excepciones informadas.

Las agregaciones de las que se dispone son “Diaria” cuyos valores son horarios, “Mensual” cuyos valores son diarios, “Anual” cuyos valores son mensuales e “Interanual” cuyos valores son anuales.

### *Tipos de peticiones*

Los selectores de fecha dan al usuario la opción de crear peticiones por rangos de fecha de tal forma que si en ambos se selecciona la misma fecha se hará una petición normal mientras que si se seleccionan distintas (siempre y cuando la primera sea menor que la segunda) se creará una petición por intervalos.

### *Visualizando informes*

La aplicación permite al usuario visualizar los informes de manera estructurada. Una vez generado un informe, los datos se muestran de manera automática en forma de tabla y de gráfica. Siguiendo el mismo ejemplo usado hasta ahora, los datos se visualizarán tal y como muestra la figura 50.

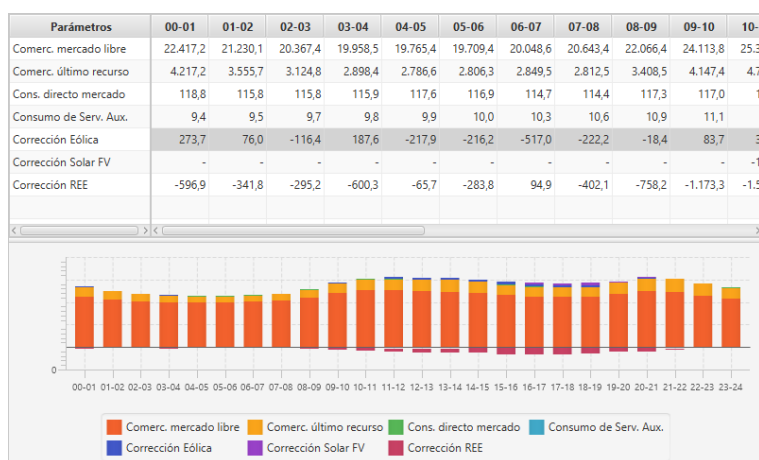


Figura 50: Visualización del informe

Es posible reorganizar las columnas de la tabla en caso de que se desee comparar dos de ellas de forma sencilla. Para acceder a todos los datos, la tabla es desplazable. En caso de necesitar más superficie, la ventana es redimensionable y las vistas de tabla y gráfica se pueden reequilibrar mediante una vista partida, haciendo más grande una vista y reduciendo la otra a elección del usuario. Para ellos simplemente hay que situar el ratón entre ambas vistas, clicar y desplazar.

Si se ha realizado una petición por rangos de fecha, en un primer momento se mostrará el primer informe de todos dando la opción de desplazarse entre ellos con las flechas situadas entre el menú de opciones y las vistas de tabla y gráfica tal y como se muestra en la figura 51.



Figura 51: Cambio de informe en petición por intervalos

## Exportando informes

Una vez generado el informe deseado, la aplicación permite exportarlo con un cuadro de diálogo estándar del sistema operativo en el que se ejecute. Para ello es necesario pulsar el botón “Exportar informe” situado junto al botón “Generar informe”.

Una vez pulsado, el usuario podrá decidir la ruta, el nombre y la extensión del archivo entre TXT y CSV siendo el archivo con formato TXT el adaptado al formato de entrada de MatLab. La ventana de guardado queda representada en la figura 52.

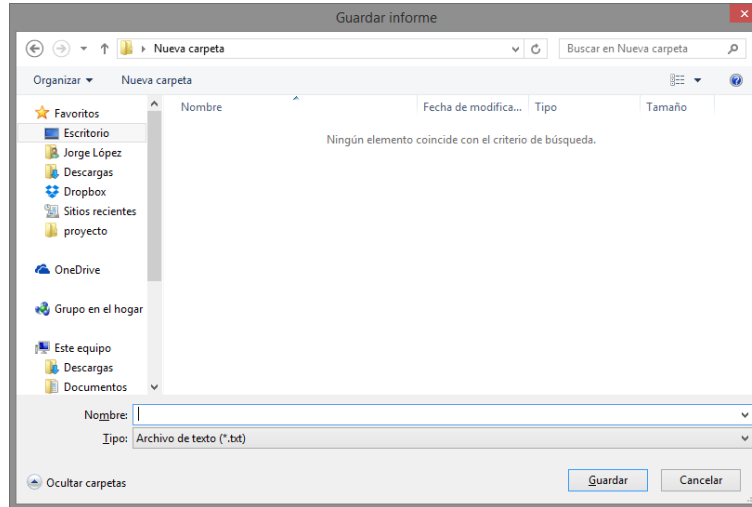


Figura 52: Pantalla de guardado del sistema

## Acerca de

La aplicación REWS ha sido creada como un proyecto universitario por Jorge Carlos López Hornero con Rosa María Romero Gómez como tutora y siendo una idea original de Julio Usaola García.



## Anexo V. Prototipo

Durante el desarrollo se ha seguido una metodología que admitía iteraciones con el objetivo de afinar al máximo dentro de las posibilidades dadas por el tiempo la interfaz final del programa, adaptándola a las necesidades del cliente y a los requisitos dados en la fase de análisis.

Antes del inicio del desarrollo se realizó un mockup para pactar con el cliente las opciones de las que debía disponer el programa mediante sencillas páginas HTML con funcionalidad acotada que mostraban la interfaz mediante imagen. En la figura 53 se aprecia una consulta de ejemplo representada en este mockup.

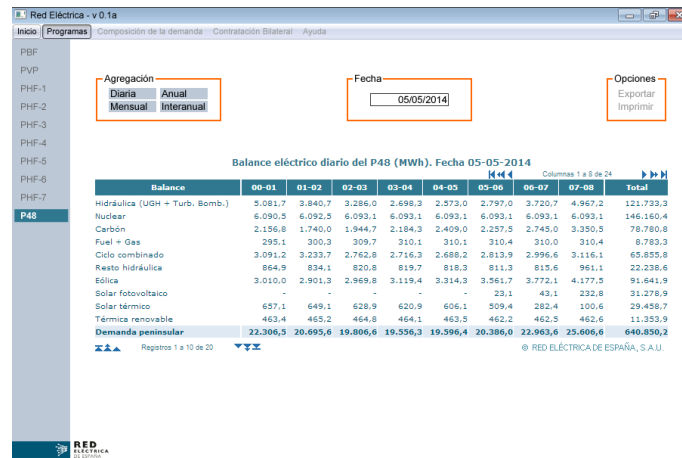


Figura 53: Ejemplo del mockup del programa

En la primera iteración, el programa no contaba con interfaz y únicamente realizaba una petición predefinida para conseguir la respuesta de los servidores de Red Eléctrica para más tarde guardarlo en un archivo HTML como salida. En la figura 54 se puede ver la salida proporcionada por esta primera toma de contacto.

Composición de la demanda diaria del PBF (MWh). Fecha 24-03-2014

Columnas 1 a 8 de 24

Demanda	00-01	01-02	02-03	03-04	04-05	05-06	06-07	07-08	Total
Comerc. mercado	18.584,0	17.687,3	16.508,2	16.319,1	16.630,9	17.552,0	19.870,9	22.878,7	553.478,4
libre	5.232,3	3.989,3	2.742,0	2.505,8	2.367,8	2.294,4	2.370,4	3.535,1	109.641,9
Comerc. último	90,4	99,7	99,8	99,7	99,9	101,1	102,3	94,9	2.431,4
recurso	2,0	1,7	1,6	1,1	1,0	1,0	1,0	1,0	27,7
Cons. directo	-	-	-	-	-	-	-	-	0,0
mercado	-	-	-	-	-	-	-	-	0,0
<b>Demanda</b>	<b>23.908,7</b>	<b>21.778,0</b>	<b>19.351,6</b>	<b>18.925,7</b>	<b>19.099,6</b>	<b>19.948,5</b>	<b>22.344,6</b>	<b>26.509,7</b>	<b>665.579,4</b>

Registros 1 a 7 de 7

Figura 54: Toma de contacto con el problema

Después de comprobar que la problemática de la obtención de datos estaba solucionada, se procedió a comenzar con la interfaz. Al ser la primera vez que se trabajaba con JavaFX el proceso de aprendizaje necesitó varias iteraciones hasta conseguir una primera versión del programa con interfaz. En la figura 55 se aprecia un primer intento de construcción de interfaz aun sin conectar con el programa.



Figura 55: Primer intento de interfaz con JavaFX

En la primera iteración con interfaz de JavaFX, el programa únicamente consistía en una muestra más o menos interactiva de cómo debía responder el programa mismo ante una petición sencilla de un solo tipo de informe, tal y como se aprecia en la figura 56. Si bien aún no mostraba el informe en la propia interfaz, sí que se guardaba en archivo HTML y dejaba introducir la fecha deseada.

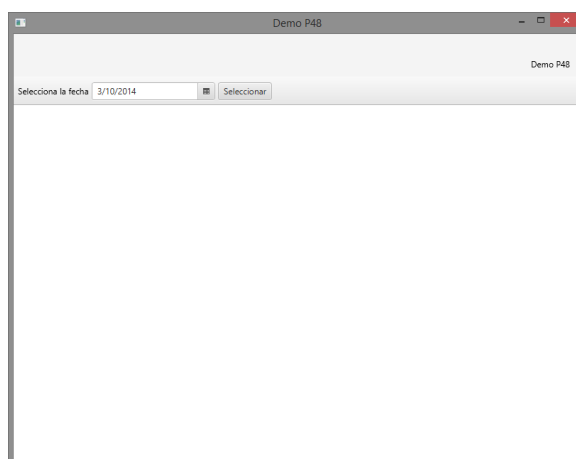


Figura 56: Primera interfaz semifuncional

La siguiente iteración contaba ya con un menú acotado que permitía interactuar con las distintas partes de la aplicación disponibles. El menú contaba con todos los informes disponibles. La vista de bienvenida se introdujo en esta iteración aunque simplemente contaba con el logo de Red Eléctrica. Al igual que en la demo anterior, los resultados aún no se mostraban en la interfaz pero si eran procesados en lugar de simplemente guardarse como HTML. En la figura 57 se aprecia la perspectiva de respuesta de esta iteración.

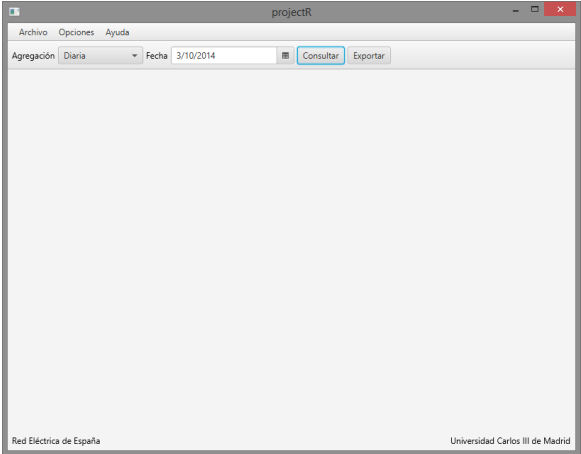


Figura 57: Primera acercamiento al resultado final

Antes de aprender a manejar las tablas y de cara a un prototipo funcional que enseñar al cliente, los resultados de la consulta se introdujeron en forma de campos de texto que eran rellenados por los controladores. La figura 58 muestra esta iteración con una consulta realizada.

The screenshot shows the 'projectR' application window with a table of electricity data. The table has columns for time intervals (00-01 to 06-07) and rows for different data categories. The 'Demanda P48' label is visible in the top right corner of the table area. The footer remains the same as in Figure 57.

	00-01	01-02	02-03	03-04	04-05	05-06	06-07
Mercado libre	21007.7	20298.3	19885.9	19448.4	19437.8	19626.4	21291.5
Último recurso							
Consumo directo	106.6	112.2	113.3	110.3	111.1	111.6	110.7
Consumo auxiliar	10.4	9.0	9.4	9.8	10.1	13.1	14.5
Corrección eólica	null	null	null	null	null	null	null
Corrección solar	null	null	null	null	null	null	null
Corrección REE	null	null	null	null	null	null	null
Total							

Figura 58: Resultados de la consulta mostrados en la interfaz

Tras la entrevista con el cliente, se añadieron nuevos objetivos como la deseabilidad de la opción de peticiones por intervalos, la cual fue añadida en la siguiente iteración. En la figura 59 se aprecia este cambio, junto a las nuevas opciones de cambio de informe.

projectR

Archivo Opciones Ayuda

Agregación Mensual Fecha 3/10/2014 4/10/2014 Consultar Exportar Demanda P48

	00-01	01-02	02-03	03-04	04-05	05-06	06-07
Mercado libre	21007.7	20298.3	19885.9	19448.4	19437.8	19626.4	21291.5
Último recurso							
Consumo directo	106.6	112.2	113.3	110.3	111.1	111.6	110.7
Consumo auxiliar	10.4	9.0	9.4	9.8	10.1	13.1	14.5
Corrección edica	null	null	null	null	null	null	null
Corrección solar	null	null	null	null	null	null	null
Corrección REE	null	null	null	null	null	null	null
Total							

Red Eléctrica de España Universidad Carlos III de Madrid

Figura 59: Añadida la posibilidad de realizar peticiones por rangos de fecha

En iteraciones posteriores se reposicionaron los botones de la interfaz y se añadió la fecha de informe mostrado para ayudar a la comprensión, tal y como se muestra en la figura 60. Adicionalmente se agregaron opciones de exportación de informes y se mejoró este aspecto internamente.

projectR

Archivo Opciones Ayuda

Agregación Diaria Fechas 25/03/2015 27/03/2015 Consultar Demanda P48

< 2015-03-28 > Exportar informe actual Exportar informe completo

	00-01	01-02	02-03	03-04	04-05	05-06	06-07
Mercado libre	22435.2	21428.5	20521.2	19938.0	19720.5	20305.0	22210.1
Último recurso							
Consumo directo	110.0	111.4	111.1	77.9	77.3	76.5	77.2
Consumo auxiliar	1.4	1.5	1.5	1.5	1.7	1.9	2.2
Corrección edica	null	null	null	null	null	null	null
Corrección solar	null	null	null	null	null	null	null
Corrección REE	null	null	null	null	null	null	null
Total							

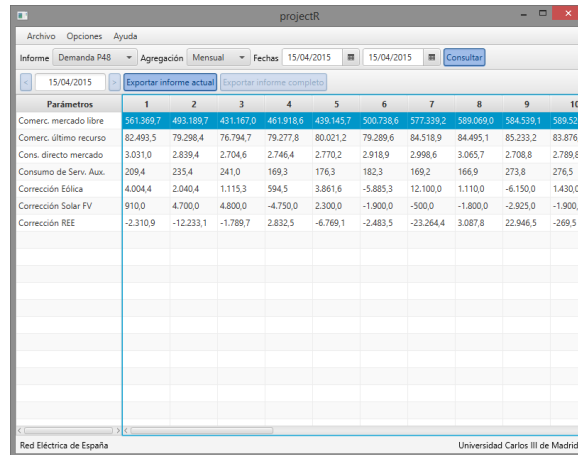
Red Eléctrica de España Universidad Carlos III de Madrid

Figura 60: Recolocación de los botones y adición de la fecha actual

En las siguientes iteraciones se realizaron cambios internos y se definieron los informes que formarían parte del programa final así como los tipos de archivos disponibles para su exportación. Se mejoraron todos estos aspectos arreglando errores y terminando temas pendientes como la posibilidad de realizar peticiones de distinto tipo y agregación. Para ello se decidió realizar una vista genérica en lugar de realizar una vista por cada tipo de petición. A nivel interno se realizó una gran reestructuración para restar complejidad computacional y redundancia de tal forma que todo tipo de petición y respuesta pudiera ser encapsulada en un solo objeto de Java en lugar de existir distintos tipos.

A nivel grafico la siguiente iteración relevante corresponde a la sustitución de los campos de texto por dos tablas en las que mostrar de manera sencilla y visual los informes, correspondiendo la de la izquierda a los parámetros y la de la derecha a los datos. En la figura 61 se aprecia esta tabla

además de la adición del selector de informe y el cambio de diseño de los botones de acción. En esta iteración también se sustituyó el logo de Red Eléctrica de la vista de bienvenida por el de la Universidad Carlos III.



The screenshot shows the 'projectR' application window. The 'Informe' dropdown is set to 'Demanda P48', 'Agregación' is 'Mensual', and 'Fechas' are '15/04/2015' to '15/04/2015'. The 'Consultar' button is visible. Below the controls is a table with 10 columns (1-10) and several rows of data. The rows include 'Comerc. mercado libre', 'Comerc. último recurso', 'Cons. directo mercado', 'Consumo de Serv. Aux.', 'Corrección Edicia', 'Corrección Solar FV', and 'Corrección REE'. The bottom status bar shows 'Red Eléctrica de España' and 'Universidad Carlos III de Madrid'.

Parámetros	1	2	3	4	5	6	7	8	9	10
Comerc. mercado libre	561.369,7	493.109,7	431.167,0	461.918,6	439.145,7	500.738,6	577.339,2	589.069,0	584.539,1	589.520,0
Comerc. último recurso	82.493,5	79.298,4	76.794,7	79.277,8	80.021,2	79.289,6	84.518,9	84.495,1	85.233,2	83.876,8
Cons. directo mercado	3.031,0	2.839,4	2.704,6	2.746,4	2.770,2	2.918,9	2.998,6	3.065,7	2.708,8	2.789,8
Consumo de Serv. Aux.	209,4	235,4	241,0	169,3	176,3	182,3	169,2	166,9	273,8	276,5
Corrección Edicia	4.004,4	2.040,4	1.115,3	594,5	3.861,6	-5.895,3	12.100,0	1.110,0	-6.150,0	1.430,0
Corrección Solar FV	910,0	4.700,0	4.800,0	-4.750,0	2.300,0	-1.900,0	-500,0	-1.800,0	-2.925,0	-1.900,0
Corrección REE	-2.310,9	-12.233,1	-1.789,7	2.832,5	-6.769,1	-2.483,5	-23.264,4	3.087,8	22.946,5	-269,5

Figura 61: Cambio de los campos de texto por tablas y otras mejoras

Al implementar distintos informes, la extensión vertical de estos por acumulación de parámetros hizo necesaria la búsqueda de una solución para la sincronización del desplazamiento entre las dos tablas de la vista. Este y otros detalles se explican al final de este anexo.

Para continuar arreglando pequeños detalles, se implementó el nombre correcto de las columnas de los datos obteniéndolos directamente de la respuesta del informe en lugar de estar predefinidos. El siguiente cambio significativo que se realizó fue añadir una gráfica de barras apiladas en la parte inferior de la perspectiva de respuesta que contiene las barras generadas por los datos del informe. En la figura 62 se pueden apreciar estos dos cambios además de la eliminación de la opción de exportar el informe actual debido a su innecesaridad.

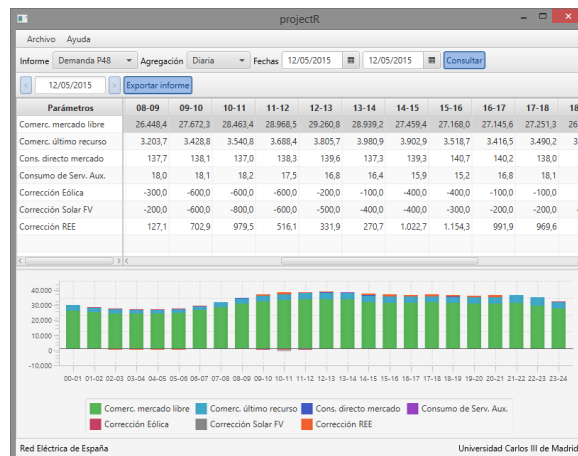
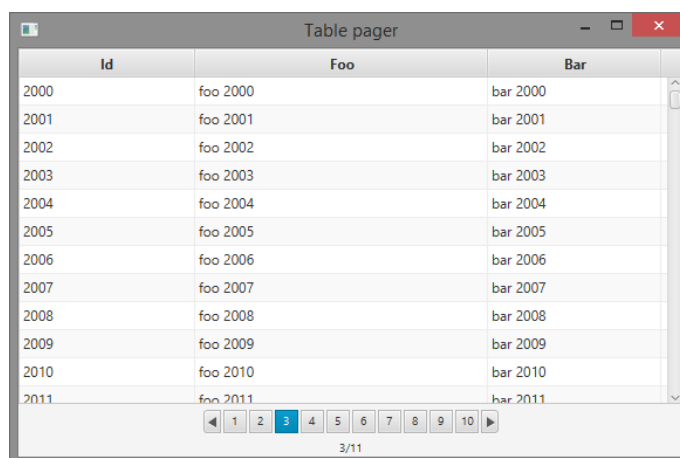


Figura 62: Añadida la gráfica y otras mejoras

De cara a la versión final se cambió la pantalla de bienvenida con un diagrama explicativo del funcionamiento y un breve texto de explicación para poder comenzar a trabajar. A nivel interno se arreglaron los fallos encontrados y eliminaron las clases innecesarias redistribuyendo el código de forma más lógica. De igual manera se eliminaron del menú opciones innecesarias. En la perspectiva de bienvenida se añadió el nombre del informe generado y se aclararon algunos elementos de la interfaz además de reubicar el botón de exportar.

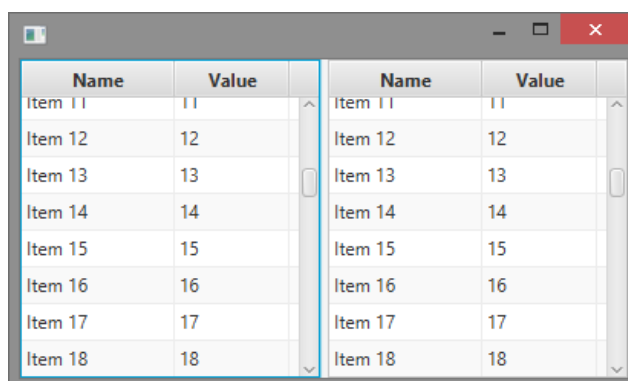
Para conseguir ciertos aspectos de la interfaz se han usado ejemplos de código disponibles en internet con ejemplos como la paginación o la creación de gráficas, procedentes entre otras fuentes de la documentación oficial de Oracle. En la figura 63 se muestra un ejemplo de paginación desechado para el desarrollo final, que iba a ser usado para mostrar los informes que se exceden del ancho de la interfaz. Finalmente se optó por implementar una clásica tabla con desplazamiento horizontal.



Id	Foo	Bar
2000	foo 2000	bar 2000
2001	foo 2001	bar 2001
2002	foo 2002	bar 2002
2003	foo 2003	bar 2003
2004	foo 2004	bar 2004
2005	foo 2005	bar 2005
2006	foo 2006	bar 2006
2007	foo 2007	bar 2007
2008	foo 2008	bar 2008
2009	foo 2009	bar 2009
2010	foo 2010	bar 2010
2011	foo 2011	bar 2011

Figura 63: Ejemplo de paginación

En la figura 64 se muestra un ejemplo del programa que ayudó a conseguir la sincronización de desplazamiento entre las distintas tablas de la interfaz.



Name	Value
Item 11	11
Item 12	12
Item 13	13
Item 14	14
Item 15	15
Item 16	16
Item 17	17
Item 18	18

Name	Value
Item 11	11
Item 12	12
Item 13	13
Item 14	14
Item 15	15
Item 16	16
Item 17	17
Item 18	18

Figura 64: Ejemplo de sincronización de desplazamiento vertical entre tablas

En la figura 65 se muestra un ejemplo de la documentación de Oracle de la creación de una gráfica con barras apiladas como la usada en la aplicación.

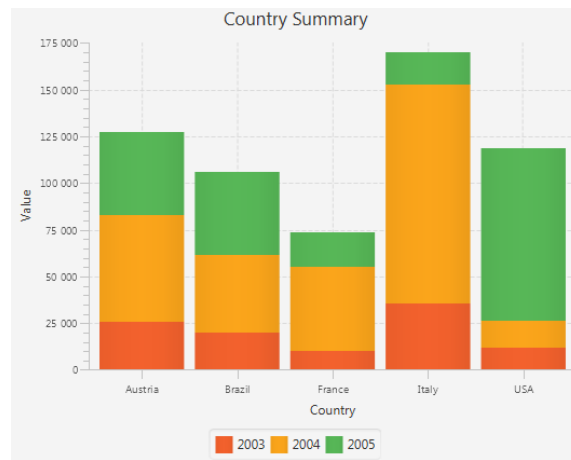


Figura 65: Ejemplo de Oracle de gráfica de barras apiladas

## Anexo VI. Formulario de usabilidad

### Formulario de usabilidad de la aplicación REWS

Este formulario busca analizar la usabilidad de la aplicación en el contexto del usuario objetivo, es decir, se debe responder poniéndose en el lugar del usuario que realmente sería proclive a usar la aplicación dadas sus necesidades y que encontraría beneficio alguno en forma de tiempo o procesamiento de datos usando la aplicación en lugar de la web a la que sustituye.

Puedes descargar la aplicación para Windows desde este enlace:

<https://www.dropbox.com/s/eeug38b5lw5es0b/REWS-0.4.exe?dl=0>

Aplicación para cualquier sistema operativo:

<https://www.dropbox.com/s/1jw14azzbsoufgg/REWS-0.4.zip?dl=0>

Para comenzar a utilizarla ejecutar REWS.jar/REWS.exe

**\*Obligatorio**

#### Experiencia \*

Introduce tu experiencia en este campo por favor

- ☐ Menos de un año  
☐ Entre uno y tres años  
☐ Más de tres años

#### Género \*

Introduce tu género por favor

- ☐ Femenino  
☐ Masculino

#### 1 - Creo que usaré esta aplicación con frecuencia \*

1 2 3 4 5

En completo desacuerdo ☐ ☐ ☐ ☐ ☐ Completamente de acuerdo

#### 2 - Encontré la aplicación innecesariamente compleja \*

1 2 3 4 5

En completo desacuerdo ☐ ☐ ☐ ☐ ☐ Completamente de acuerdo

#### 3 - Pienso que la aplicación es fácil de usar \*

1 2 3 4 5

En completo desacuerdo ☐ ☐ ☐ ☐ ☐ Completamente de acuerdo



**4 - Creo que necesitaría el apoyo de un experto para usar la aplicación \***

1 2 3 4 5

En completo desacuerdo ☐ ☐ ☐ ☐ ☐ Completamente de acuerdo

**5 - Encontré las diversas posibilidades de la aplicación bien integradas \***

1 2 3 4 5

En completo desacuerdo ☐ ☐ ☐ ☐ ☐ Completamente de acuerdo

**6 - Pienso que hay demasiada inconsistencia en la aplicación \***

1 2 3 4 5

En completo desacuerdo ☐ ☐ ☐ ☐ ☐ Completamente de acuerdo

**7 - Imagino que la mayoría de las personas aprenderían muy rápidamente a utilizar la aplicación \***

1 2 3 4 5

En completo desacuerdo ☐ ☐ ☐ ☐ ☐ Completamente de acuerdo

**8 - Encontré la aplicación muy grande al recorrerla \***

1 2 3 4 5

En completo desacuerdo ☐ ☐ ☐ ☐ ☐ Completamente de acuerdo

**9 - Me sentí muy confiado en el manejo de la aplicación \***

1 2 3 4 5


En completo desacuerdo ☐ ☐ ☐ ☐ ☐ Completamente de acuerdo

**10 - Necesito aprender muchas cosas antes de manejarla con la aplicación \***

1 2 3 4 5

En completo desacuerdo ☐ ☐ ☐ ☐ ☐ Completamente de acuerdo

**Enviar**

Con la tecnología de  Google Forms

Este contenido no ha sido creado ni aprobado por Google.  
[Informar sobre abusos](#) - [Condiciones del servicio](#) - [Otros términos](#)

Figura 66: Formulario de usabilidad de la aplicación